(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2000－75785
（P2000－75785A）
(43)公開日　平成12年３月14日(2000.3.14)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード（参考）
Ｇ０９Ｃ　1/00	６１０	Ｇ０９Ｃ　1/00	６１０Ｂ	
Ｈ０４Ｌ　9/06		Ｈ０４Ｌ　9/00	６１１Ａ	

審査請求　未請求　請求項の数15　ＯＬ　（全 20 頁）

(21)出願番号　特願平10－240844

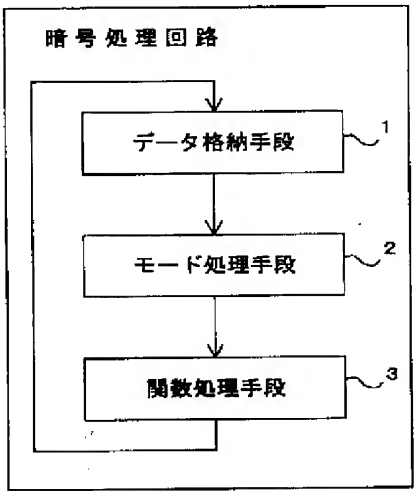
(22)出願日　平成10年８月26日(1998.8.26)

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(72)発明者　長谷部　高行
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　岡田　壮一
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(74)代理人　100074099
弁理士　大菅　義之　（外１名）

(54)【発明の名称】　高速暗号処理回路および処理方法

(57)【要約】
【課題】　ＤＥＳ暗号の操作モードとしてのＣＢＣモードとＣＦＢモードとの両方を実行できる暗号処理回路を提供し、かつその回路における処理遅延をできるだけ削減する。
【解決手段】　ブロック暗号化処理に用いられる初期ベクトル、処理中間値、または処理最終値を格納する手段１と、ブロック暗号化処理内で用いられる関数処理を実行する手段２と、手段１と手段２との間に備えられ、ブロック暗号化の操作モードに対応する処理を実行する手段３とを備える。

第１の実施形態に対応する原理構成ブロック図



1

【特許請求の範囲】

【請求項１】　ブロック暗号化の処理を行う暗号処理回路において、

ブロック暗号化処理に用いられる初期ベクトル、処理中間値、または処理最終結果を格納するデータ格納手段と、

ブロック暗号化処理内で用いられる関数処理を実行する関数処理手段と、

該データ格納手段と関数処理手段との間に備えられ、ブロック暗号化の操作モードに対応する処理を実行するモード処理手段とを備えることを特徴とする高速暗号処理回路。

【請求項２】　前記モード処理手段が排他的論理和回路によって構成されることを特徴とする請求項１記載の高速暗号処理回路。

【請求項３】　前記ブロック暗号がＤＥＳ暗号であり、前記操作モードがＣＦＢモードであることを特徴とする請求項１、または２記載の高速暗号処理回路。

【請求項４】　前記関数処理手段が、前記ＤＥＳ暗号へのブロックデータの暗号化における前記関数処理としての非線形関数を用いる１６段の変換を複数回に分けて実行することを特徴とする請求項３記載の高速暗号処理回路。

【請求項５】　前記関数処理手段が、前記複数回の最後の１回を除く毎回の実行結果を前記データ格納手段に出力することを特徴とする請求項４記載の高速暗号処理回路。

【請求項６】　前記関数処理手段による、前記複数回の最終回の処理結果を用いて、入力ブロックデータに対する暗号化ブロックデータを出力する暗号化データ出力手段を更に備えることを特徴とする請求項４記載の高速暗号処理回路。

【請求項７】　前記関数処理手段による、前記複数回の最初の回の実行の前に、前記モード処理手段がモード処理を実行し、該実行結果を該関数処理手段に出力すると共に、

該モード処理の実行結果の一部を格納するモード処理結果格納手段を更に備えることを特徴とする請求項４記載の高速暗号処理回路。

【請求項８】　前記関数処理手段による、前記複数回の最後の回の実行時に、前記モード処理結果格納手段が保持しているデータを前記データ格納手段に出力すると共に、該関数処理手段が該実行結果の一部を該データ格納手段に出力することを特徴とする請求項７記載の高速暗号処理回路。

【請求項９】　前記ブロック暗号がＤＥＳ暗号であり、前記操作モードがＣＢＣモードであることを特徴とする請求項１、または２記載の高速暗号処理回路。

【請求項１０】　前記関数処理手段が、前記ＤＥＳ暗号へのブロックデータの暗号化における前記関数処理としての非線形関数を用いる１６段の変換を複数回に分けて実行することを特徴とする請求項９記載の高速暗号処理回路。

2

【請求項１１】　前記関数処理手段が、前記複数回の毎回の実行結果を前記データ格納手段に出力することを特徴とする請求項９記載の高速暗号処理回路。

【請求項１２】　前記関数処理手段による前記複数回の最初の回の実行の前に、前記モード処理手段が入力されるブロックデータと前記格納手段に格納されているデータとを用いてモード処理を実行し、該モード処理の結果を該関数処理手段に出力することを特徴とする請求項９記載の高速暗号処理回路。

【請求項１３】　ブロック暗号化の処理を行う暗号処理回路において、

ブロック暗号化処理に用いられる初期ベクトル、処理中間値、または処理最終結果を格納するデータ格納手段と、

ブロック暗号化処理内で用いられる関数処理を実行する関数処理手段と、

該データ格納手段と関数処理との間に備えられ、ブロック暗号化の操作モードに対応する第１の処理を実行する第１のモード処理手段と、

該関数処理手段とデータ格納手段との間に備えられ、ブロック暗号化の操作モードに対応する第２の処理を実行する第２のモード処理手段とを備えることを特徴とする高速暗号処理回路。

【請求項１４】　ブロック暗号化の処理を行う暗号処理方法において、

１つのブロックデータの暗号化処理の終了時点で処理結果を出力すると共に、初期ベクトルを格納していたレジスタに該暗号化処理における処理中間値、または処理最終結果を格納し、

該１つのブロックデータ、または該１つのブロックに続く次のブロックのデータとレジスタに格納されたデータとの間で該ブロック暗号化の操作モードに対応する演算を実行し、

該演算結果を、該１つのブロックに続く次のブロックデータに対する暗号化処理における前記初期ベクトルに相当する値として用いて、該次のブロックデータに対する暗号化処理を行うことを特徴とする暗号処理方法。

【請求項１５】　前記ブロック暗号がＤＥＳ暗号であり、前記操作モードがＣＦＢモード、たまはＣＢＣモードであることを特徴とする請求項１４記載の暗号処理方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明はディジタルデータの暗号化方式に関し、更に詳しくは代表的なブロック暗号としてのＤＥＳ暗号のＣＢＣモードとＣＦＢモードの処理を実行する高速暗号処理回路に関する。

【０００２】
【従来の技術】近年ディジタルデータの暗号化処理が注目されており、暗号化処理の対象となるデータも増加している。例えば動画像データの暗号化処理や、サーバにおけるトランザクション処理の要求が出てきており、暗号化処理自体にも高速化が求められている。
【０００３】ディジタルデータの暗号化方式としては、例えば６４ビットを１つのブロックとしてブロック毎に暗号化を行うブロック暗号方式と、例えば１ビットずつ逐次的に暗号化を行うストリーム暗号方式とがある。
【０００４】本発明が対象とする暗号化方式はブロック暗号であり、その中でも最も代表的なＤＥＳ暗号である。ＤＥＳ（データ　エンクリプション　スタンダード）は1977年１月にアメリカ商務省標準局によって定められたデータ暗号標準であり、現在最も広範に用いられている暗号アルゴリズムである。
【０００５】ＤＥＳ暗号に対しては、操作モードとして４つのモードが定められている。そのモードはＥＣＢ（エレクトロニック　コード　ブロック），ＣＢＣ（サイファブロック　チェイニング），ＣＦＢ（サイファ　フィードバック）およびＯＦＢ（アウトプット　フィード　バック）の４つである。
【０００６】これらの４つのモードのうち、本発明に直接関連があるのはＣＢＣ、およびＣＦＢの２つのモードであるため、これらの２つの操作モードにおける暗号化および復号の処理について、図２４～図２７を用いて説明する。
【０００７】図２４はＣＢＣモードにおける暗号化および復号の基本説明図である。このモードでは６４ビットに分割された平文系列M_1，M_2，M_3，・・・が入力されるが、まず最初の６４ビットとしてのM_1に対しては初期ベクトルＩＶとの排他的論理和５１がとられ、その結果に対して暗号化処理Ｅ５２が実行されて暗号文系列C_1，C_2，C_3，・・・のうちの最初の６４ビットとしてのC_1が得られる。ここで暗号化処理としてのＥ５２は初期転置ＩＰ，ｆ関数を用いた同一構造の１６段の変換、左右３２ビットの入替え、および逆転置IP^{-1}を含むものである。
【０００８】次の６４ビットとしてのM_2の入力に対しては、暗号文の最初の６４ビットとしてのC_1との排他的論理和５６がとられ、その結果に対して暗号化処理Ｅ５７が行われ、暗号文の次の６４ビットとしてのC_2がられる。以下同様にして暗号文が６４ビットずつ作成される。
【０００９】復号処理においては、暗号文の最初の６４ビットとしてのC_1の入力に対して復号処理Ｄ５３が実行される。この処理は暗号化処理Ｅ５２と同様に初期転置ＩＰ，ｆ関数を用いた１６段の変換、左右３２ビットの入替え、および逆転置IP^{-1}を含むものであるが、暗号化処理と異なって１６段の変換においてキー系列が逆の順序で用いられる。復号処理Ｄ５３の結果に対して、初期ベクトルＩＶとの排他的論理和５４がとられ、平文の最初の６４ビットとしてのM_1が得られる。
【００１０】暗号文の次の６４ビットとしてのC_2の入力に対して復号処理Ｄ５８が行われた後に、その結果に対して暗号文の最初の６４ビットとしてのC_1との排他的論理和５９がとられ、平文の次の６４ビットとしてのM_2が得られる。以下同様にして復号結果としての平文が作成される。
【００１１】図２５はＣＦＢモードの動作の基本説明図である。ＣＦＢモードは乱数発生動作を行うものであり、平文は１以上６４以下であるｋビットずつに分割され、その系列は図２４におけると同様にM_1，M_2，M_3，・・・とされる。
【００１２】平文の、例えば最初の８ビットとしてのM_1の入力に対応して、レジスタ６６に格納されている初期ベクトルＩＶの暗号化処理Ｅ６７が行われる。この暗号化処理そのものは図２４のＣＢＣモードにおけると同じである。暗号化処理の結果の６４ビットのうち左側の８ビットだけが取り出され、M_1との排他的論理和６８がとれら、暗号文の最初の８ビットC_1が出力されると共に、その暗号文としての８ビットC_1はレジスタ６６の最も右側に格納される。すなわちレジスタ６６に格納されていた初期ベクトルＩＶは８ビット左シフトされて、左側の８ビットが捨てられ、最も右側にC_1が格納されることになる。
【００１３】平文の次の８ビットM_2の入力に対応して、レジスタ６６の格納内容に対する暗号化処理Ｅ６７が行われ、その結果の６４ビットのうちの左８ビットとM_2との排他的論理和６８がとれら、暗号文の次の８ビットC_2として出力されると共に、その８ビットはレジスタ６６の最も右側に格納される。すなわちレジスタ６６の格納内容は８ビット左シフトされ、初期ベクトルＩＶのうち更に８ビットが捨てられて、レジスタ６６の最も右側にC_2が格納されることになる。暗号文系列の次の８ビットC_3以下については全く同様の動作が行われる。
【００１４】ＣＦＢモードの復号処理においては、まず暗号文の最初の８ビットC_1の入力に対応して、レジスタ７０に格納されていた初期ベクトルＩＶの暗号化処理Ｅ７１が行われる。図２４のＣＢＣモードと異なり、復号処理においても処理Ｄではなく、処理Ｅが用いられる。この処理の結果の６４ビットのうち左８ビットが取り出され、暗号文のC_1との排他的論理和７２がとられ、平文の最初の８ビットM_1が出力される。それと同時に暗号化処理におけると同様に、レジスタ７０の内容は８ビット左シフトされ、最も右側に暗号文の８ビットC_1が格納される。
【００１５】暗号文系列の次の８ビットC_2の入力に対応する処理は全く同様である。すなわちレジスタ７０の

内容に対して暗号化処理Ｅ７１が行われ、その結果の６４ビット中の左８ビットとC_2との排他的論理和７２がとられ、平文の次の８ビットM_2が得られると共に、レジスタ７０の内容は８ビット左シフトされ、暗号文の８ビットC_2がレジスタ７０の最も右側に格納される。以下同様にして、暗号文系列C_3，・・・の復号処理が実行される。

【００１６】図２６はＣＢＣモードの暗号処理回路の従来例である。同図(a)は暗号化回路を示す。同図において、図２４におけると同様に平文の最初の８ビットM_1が入力されると、レジスタ７５に格納されている初期ベクトルＩＶとの排他的論理和がＥＸＯＲ回路７６によってとられ、その結果はｆ関数７７に入力される。

【００１７】ｆ関数７７としては、ｆ関数を用いた１６段の変換を全て行うために１６段の変換回路を備えてもよいが、例えばハードウェア量を減らすために、４段の変換回路を備え、処理の中間結果をレジスタ７５およびＥＸＯＲ回路７６を介して４回ループさせることによって、１６段の変換を実現することもできる。この場合、そのループの処理の間は、ＥＸＯＲ回路７６に対する入力の値を６４ビット全てに対して例えば“０”とすることによって、レジスタ７５の格納内容をそのままｆ関数７７に出力することができる。

【００１８】なおここでは簡単のためにｆ関数７７のみを示したが、図２４の暗号化処理Ｅ５２に含まれる初期転置ＩＰは例えば第１段目のｆ関数の処理に含まれるものとし、また最後の左右３２ビットの入替え、および逆転置IP^{-1}は１６段目のｆ関数の処理に含まれるものと考えることができる。

【００１９】そして４回目のループの終了時点でｆ関数７７から出力され、レジスタ７５に格納される６４ビットは、暗号文の最初のブロックC_1として出力されると共に、入力平文の次のブロックM_2の入力時点でM_2と排他的論理和をとるために用いられる。以下の処理は図２４に対する説明と同様であるので、その説明を省略する。

【００２０】図２６(b)はＣＢＣ復号回路の従来例である。同図において暗号文の最初のブロックC_1が入力されると、その入力はセレクタ８０を介してｆ関数８１に与えられる。

【００２１】このｆ関数８１は図２４における復号処理Ｄ５３を実行するためのものであり、(a)と同様に４段の変換回路を備えるものとすると、レジスタ８２、セレクタ８０を介する４回のループの動作の終了後に入力暗号文ブロックC_1の復号結果をレジスタ８２に出力することになり、また第１段目の変換には初期転置ＩＰ、１６段目の変換には左右３２ビットの変換と逆転置IP^{-1}を含むものとする。

【００２２】レジスタ８２に格納された暗号文の最初のブロックC_1の復号結果はＥＸＯＲ回路８４に入力され、レジスタ８３に格納されていた初期ベクトルＩＶとの排他的論理和がとられ、平文の最初のブロックM_1として出力される。この時次のブロックC_2の復号結果と排他的論理和をとるために、入力暗号文の最初のブロックC_1がレジスタ８３に格納される。暗号文の次のブロックC_2の入力に対する復号処理は図２４におけると同様に実行されるので、以下その説明を省略する。

【００２３】図２７はＣＦＢモードの暗号処理回路の従来例である。同図(a)は暗号化回路、(b)は復号回路を示す。図２７(a)において平文の最初の８ビットM_1の入力に対応して、まずレジスタ８６に格納されている初期ベクトルＩＶの値がｆ関数８７に出力される。

【００２４】ｆ関数８７は図２５における暗号化処理Ｅ６７、すなわち図２４における暗号化処理Ｅ５２と同様の処理を行うものであり、図２６におけると同様に例えば４段の変換回路が備えられる。セレクタ９０およびレジスタ８６を介する４回のループの処理による１６段の変換（図２６におけると同様に初期転置、左右３２ビットの入替え、および逆転置を含む）の処理結果としての６４ビットのうち、左８ビットのみがＥＸＯＲ回路９１に出力され、入力平文の８ビットM_1と排他的論理和がとれら、暗号文の最初のブロックC_1として出力される。

【００２５】前述のように、レジスタ８６に格納されていた初期ベクトルＩＶはM_1の入力に対応してｆ関数８７に出力されるが、この時その内容は８ビット左にシフトされ、すなわち左の８ビットが捨てられて、レジスタ８８に格納される。そのビット数は５６ビットである。ｆ関数８７の４回目のループの後の出力のうち、最も左側の８ビットのみがＥＸＯＲ回路８９に入力され、入力M_1との排他的論理和がとられ、その結果の８ビットが最も右側に、レジスタ８８に格納されている５６ビットが左側に配置される形式で、セレクタ９０を介してレジスタ８６に６４ビットのデータとして格納される。この６４ビットは、平文の次のブロックM_2の入力に対応して、図２５の暗号化処理Ｅ６７に出力されることになる。M_2以下のブロックに対する動作は同様であるので、以下の説明を省略する。

【００２６】図２７(b)において、暗号文の最初の８ビットC_1の入力時点で、レジスタ９３に格納されている初期ベクトルＩＶがｆ関数９４に与えられ、前述と同様にセレクタ９５、レジスタ９３を介する４回のループの処理によって１６段のｆ関数を用いる変換などが行われ、図２５の暗号化処理Ｅ７１の出力としての６４ビットのうち左８ビットがＥＸＯＲ回路９７に出力され、入力暗号文ブロックC_1との排他的論理和がとられ、平文M_1が出力される。

【００２７】レジスタ９３の内容としての初期ベクトルＩＶの値がｆ関数９４に出力される時点で、同時にその内容は８ビット左シフトされ、５６ビットがレジスタ９

６に格納される。そして４回目のループの終了時点でその５６ビットが左側、入力暗号文８ビットC_1が右側に配置される形式で、セレクタ９５を介してレジスタ９３に６４ビットのデータが格納される。このデータは、次の暗号文ブロックC_2の入力に対応して、ｆ関数９４に与えられるものである。暗号文ブロックC_2以下に対する復号処理については同様であるので、その説明を省略する。

【００２８】

【発明が解決しようとする課題】このようなＤＥＳ暗号化処理を高速化するために、従来は特にｆ関数を用いた１６段の変換処理の高速化を目的として、ｆ関数の変換回路をできるだけ段数多く接続してループ損を削減したり、ｆ関数を含む変換の処理手順を変更して１回のループに対する遅延を削減したり、例えばＣＢＣ暗号化処理において操作モードに対応するモード処理を暗号化の処理サイクル、すなわちループの中に入れて処理に必要なクロック数を減らすというような方式が用いられていた。これらの方式に関しては、次の米国特許に開示されている。

【００２９】Ｕ．Ｓ．Patent 5,381,480, Butter et al. "System for Translating Encrypted Data"

しかしながらｆ関数を用いる変換をできるだけ段数多く接続するとしても、段数を多くすればするほどハードウェア量が増大し、コストが大きくなるために、あまり段数を多くすることもできないという問題点があった。

【００３０】次にＤＥＳブロック暗号の操作モードとしては前述のようにＥＣＢ，ＣＢＣ，ＣＦＢ，ＯＦＢの４つの操作モードが存在する。ＥＣＢモードはＤＥＳ暗号化処理と復号処理とを組み合わせるだけであり、操作モードのための特別なモード処理回路は不必要である。ＯＦＢモードにおいては、ＤＥＳ暗号化の処理ループ内では、前のブロックに対する処理結果を用いてそのまま次のブロックに対する処理を行うことができ、モードに対応する特別な処理としてはＤＥＳの最終処理結果と入力ブロックデータとの間で排他的論理和をとるだけでよく、処理ループに対しては遅延の発生を考慮しなくてよい。

【００３１】これに対してＣＢＣモードでは、例えば初期ベクトルＩＶの値と入力ブロックとの排他的論理和をとり、この値に対してＤＥＳ処理を行う必要がある。またＣＦＢモードでは、暗号化処理において初期ベクトルＩＶを格納していたレジスタの内容更新時において、初期ベクトルの値の左シフトを行い、右側にＤＥＳ処理の処理結果の一部と入力ブロックとの排他的論理和の値を設定する必要がある。

【００３２】すなわちＣＢＣに対するモード処理はＤＥＳ処理の最初で、またＣＦＢに対するモード処理は最後で行われるという違いがある。これらの処理は共にＤＥＳ処理のループの中に含まれている。

【００３３】ＤＥＳ暗号化の処理を行う処理回路としては、これらの４つの操作モードの全てを実行できる回路を構成することが望ましいが、従来においては特にＣＢＣモードとＣＦＢモードとに共通に用いられる処理回路を構成することが困難であるという問題点があった。これはこれらの２つのモードに対応するモード処理が一方はＤＥＳ処理の処理ループの最初、他方は最後に入っていることに起因する。

【００３４】本発明は特にＣＢＣモードとＣＦＢモードの両方を実行できる暗号処理回路を提供することと、その回路における処理遅延をできるだけ削減することを目的とする。

【００３５】

【課題を解決するための手段】図１は本発明の第１の実施形態に対応する原理構成ブロック図である。同図は、例えば操作モードとしてＣＢＣモードと、ＣＦＢモードとの両方のＤＥＳ暗号化処理を実行できる、ブロック暗号化の処理を行う高速暗号処理回路の構成ブロック図である。

【００３６】図１において、データ格納手段１はブロック暗号化処理に用いられる初期ベクトル、処理中間値、または処理最終結果を格納するものであり、例えばレジスタである。

【００３７】関数処理手段２はブロック暗号化処理、例えばＤＥＳ処理内で用いられる関数処理としてのｆ関数を用いる複数段の変換を実行するものである。モード処理手段３はデータ格納手段１と関数処理手段２との間に備えられ、データ格納手段１に格納されているデータを用いてブロック暗号化の操作モード、例えばＣＢＣモードまたはＣＦＢモードに対応する処理を実行するものである。

【００３８】ＣＦＢモードの処理において、関数処理手段２によって、ＤＥＳ暗号への入力ブロックデータの暗号化における前述の関数処理としての非線形関数を用いる１６段の変換が複数回、例えば４回に分けて実行される。その場合には１回の関数処理においては、非線形関数を用いる４段の変換が実行される。

【００３９】この複数回の処理の最後の１回を除く毎回の実行結果は、関数処理手段２によってデータ格納手段１に格納される。そしてこの最後の１回の処理結果を用いて入力ブロックデータに対する暗号化ブロックデータを出力するために、暗号化データ出力手段が更に備えられる。

【００４０】またこの複数回の処理の最初の回の実行の前に、モード処理手段３によってモード処理が実行され、その実行結果が関数処理手段２に出力されると共に、モード処理の実行結果の一部が格納されるモード処理結果格納手段が更に備えられる。そして関数処理手段２による複数回の処理の最後の回の実行時に、モード処理結果格納手段によって前述のモード処理の実行結果の

一部がデータ格納手段１に出力されると共に、関数処理手段２によって実行結果の一部がデータ格納手段１に出力される。

【００４１】ＣＢＣモードの暗号化処理の実行時においては、ＣＦＢモードにおけると同様に、前述のＤＥＳ暗号へのブロックデータの暗号化における関数処理としての非線形関数を用いる１６段の変換が、複数回に分けて実行される。そしてこの複数回の処理の毎回の実行結果は、関数処理手段２によってデータ格納手段１に格納される。

【００４２】また関数処理手段２による複数回の処理の最初の回の実行の前に、モード処理手段３によって入力されるブロックデータとデータ格納手段に格納されているデータとを用いたモード処理が実行され、そのモード処理の結果が関数処理手段２に出力される。

【００４３】図２は本発明の第２の実施形態に対応する原理構成ブロック図である。同図においては図１のモード処理手段３に対応する第１のモード処理手段４に加えて、関数処理手段２とデータ格納手段１との間に備えられ、ブロック暗号化の操作モードに対応する第２の処理を実行する第２のモード処理手段５が追加される。

【００４４】本発明の第２の実施形態においては、第１の実施形態におけると同様に、暗号処理回路はＣＦＢモードとＣＢＣモードとの両方の操作モードを実行できるものであるが、ＣＢＣモードにおけるモード処理の基本的な部分を実行するのは第１のモード処理手段４であるのに対して、ＣＦＢモードにおける基本的なモード処理は第２のモード処理手段５によって実行される。

【００４５】ＣＦＢモードにおいて、関数処理手段２による非線形関数を用いる１６段の変換は、第１の実施形態におけると同様に複数回に分けて実行されるが、その最初の回においてデータ格納手段１に格納されているデータは、第１のモード処理手段３による特別な処理が行われることなく、そのまま関数処理手段２に与えられる。関数処理手段２による複数回の処理の最後の回を除く毎回の処理結果はデータ格納手段１に格納される。

【００４６】関数処理手段２による最終回の処理結果を用いて、入力ブロックデータに対する暗号化ブロックデータが第１の実施形態におけると同様に出力されると共に、その処理結果の一部と入力データブロックを用いて第２のモード処理手段によって第２のモード処理が実行され、その結果と、複数回の最初の回の実行の前にデータ格納手段１に格納されていたデータの一部とがデータ格納手段１に格納され、次のブロックデータの暗号化処理のために用いられる。

【００４７】ＣＢＣモードにおいては、第１のモード処理手段４によって第１の実施形態に対するモード処理手段３によると同様の処理が実行され、第２のモード処理手段は実質的に何らの作用を行うこともなく、結果的に第１の実施形態におけると同様の処理が実行される。

【００４８】本発明の暗号処理方法においては、例えばＤＥＳブロック暗号化の処理を行う暗号処理方法において、１つのブロックデータの暗号化処理の終了時点で処理結果が出力されると共に、初期ベクトルが格納されていたレジスタにその暗号化処理における処理中間値、または処理最終結果が格納される。

【００４９】そしてレジスタに格納されたデータとその１つのブロックデータ、またはその１つのブロックに続く次のブロックデータとの間で、そのブロック暗号化の操作モードに対応する演算が実行される。更にその演算結果が、その１つのブロックに続く次のブロックデータに対する暗号化処理における初期ベクトルに相当する値として用いられ、次のブロックデータに対する暗号化処理が行われる。

【００５０】本発明の暗号化処理方法においては、前述のレジスタに最初に初期ベクトルが格納されており、まずその初期ベクトルに対しては例えばＣＢＣモードでは入力ブロックデータとの間で操作モードに対応する演算が実行された後に、ＣＦＢモードでは操作モードに対応する特別な演算は実質的に実行されることなく、演算結果に対応して非線形関数を用いる１６段の変換が複数回に分けて実行される。この暗号化処理の方法は、前述の第１の実施形態において用いられるものである。

【００５１】以上説明したように、本発明によればＣＢＣモードとＣＦＢモードとの両方を実行する暗号処理回路を実現することができる。また第１の実施形態においては２つのモードに対応するモード処理手段を共通化することが可能になる。

【００５２】

【発明の実施の形態】図３は本発明の第１の実施形態において用いられるＣＦＢモードの暗号化回路の構成を示す。同図は図２７(a) に示した従来例に対応するものであるが、ＣＢＣモードの暗号化回路と組み合わせて使用するために図２７(a) において用いられていたＥＸＯＲ回路８９が省略され、その代わりにレジスタ１１の出力側にＥＸＯＲ回路１２が用いられている。

【００５３】図３において初期ベクトルＩＶはレジスタ１１に格納されている。平文の最初の８ビットM_1 の入力時点で、レジスタ１１から初期ベクトルＩＶの値がｆ関数（処理回路）１３に与えられる。この時ＥＸＯＲ回路１２に対してレジスタ１４から、例えば“０”が出力されることにより、レジスタ１１に格納されていた初期ベクトルＩＶの値はそのままｆ関数１３に与えられる。

【００５４】ｆ関数１３は、図２７(a) におけると同様に例えば４段のｆ関数を用いる変換回路からなり、セレクタ１５、レジスタ１１を介する４回のループの処理によってｆ関数を用いた１６段の変換が行われるものとする。また前述のように初期転置ＩＰ、第１６段目の結果に対する左右３２ビットの入替え、逆転置IP^{-1}を含むものとする。

【００５５】４回のループの後のｆ関数１３の出力のうち左８ビットはＥＸＯＲ回路１８に与えられ、入力平文ブロックM_1 との排他的論理和がとられ、最初の暗号文ブロックC_1 として出力される。

【００５６】ブロックM_1 の入力に対応してレジスタ１１から初期ベクトルＩＶが出力される時点で、その内容のうち、左側５６ビットはそのままレジスタ１１から、また右側８ビットはＥＸＯＲ回路１２を介して与えられる形式でレジスタ１７側に出力されるが、その６４ビットは８ビットシフトされて最も左側の８ビットが捨てられた後、レジスタ１７に格納される。そして４回のループの終了時点でその内容の５６ビットが左側に、ｆ関数１３の出力のうちの左８ビットが右側に格納される形式で、セレクタ１５を介してレジスタ１１に６４ビットのデータが格納される。またこの時点でレジスタ１４には入力ブロックM_1 ８ビットが格納される。ここでレジスタ１１に格納されている６４ビットは、そのままでは次の入力ブロックM_2 の入力時点でｆ関数１３に対してＤＥＳ暗号化処理の対象として用いることはできない。

【００５７】ブロックM_2 の入力時点で、レジスタ１１の格納内容のうち右側８ビットはＥＸＯＲ回路１２に出力され、レジスタ１４に格納されている最初のブロックM_1の８ビットとの排他的論理和がとられ、レジスタ１１からそのまま出力される左側５６ビットと共にｆ関数１３に入力される。この時実質的にｆ関数１３への入力となる６４ビットは、前述と同様にして左に８ビットシフトされ、５６ビットがレジスタ１７に格納される。これは次のブロックM_3 の入力時点におけるｆ関数１３への入力データ作成のためである。

【００５８】ｆ関数１３、セレクタ１５、レジスタ１１を介する４回のループの終了時点で、ｆ関数１３の出力のうち左８ビットと入力M_2 ８ビットとの排他的論理和がＥＸＯＲ回路１８によってとられ、暗号文系列の次のブロックC_2 として出力される。この４回のループ処理の間はレジスタ１４の格納内容は出力されず、ＥＸＯＲ回路１２には例えば“０”が与えられるものとする。以下同様の処理が行われるため、その説明を省略する。

【００５９】図４は本発明の第１の実施形態としての暗号処理回路の構成ブロック図である。同図はＣＦＢモードとＣＢＣモードとの両方のモードの処理を実行する暗号処理回路の構成ブロック図である。この回路を用いたＣＦＢモード暗号化動作について、図５～図９を用いて説明する。

【００６０】図５はＣＦＢモード暗号化動作（その１）の説明図である。ここで使用される回路の結線部分については実線で、また使用されない部分については破線で示してある。まず最初の平文ブロックM_1 の入力（①から）に対応して、レジスタ２１に格納されている初期ベクトルＩＶがセレクタ２２，２３、ＥＸＯＲ回路２４，２５を介してｆ関数２６に与えらる。この時セレクタ２２，２３はレジスタ２１からの出力を選択し、またＥＸＯＲ回路２４，２５に対してはセレクタ３１，３２を介して５６ビットと８ビットの“０”のデータが入力されている。このためレジスタ２１に格納されていた初期ベクトルＩＶの値はそのままｆ関数２６に与えられる。またレジスタ２１に格納されていたＩＶの値は８ビットシフトされるだけで、実質的にそのままレジスタ２８に格納される。そしてｆ関数２６に含まれる４段の変換回路の処理結果はセレクタ２７を介してレジスタ２１に格納される。

【００６１】図６はＣＦＢモード暗号化動作（その２）の説明図である。同図はｆ関数２６、セレクタ２７、レジスタ２１を介する４回のループのうち、第２回目と第３回目のループにおける処理動作の説明図である。これらのループにおいてはセレクタ２２，２３はレジスタ２１からの出力を選択し、またセレクタ３１，３２は５６ビットの“０”と８ビットの“０”をそれぞれＥＸＯＲ回路２４，２５に出力しているため、レジスタ２１のデータはそのままｆ関数２６に与えられ、４段の処理結果はセレクタ２７を介してレジスタ２１に格納される。

【００６２】図７はＣＦＢモード暗号化動作（その３）の説明図である。同図において、レジスタ２１に格納されている３回目のループの処理結果は、前述と同様にセレクタ２２，２３、ＥＸＯＲ回路２４，２５を介してそのままｆ関数２６に与えられる。ｆ関数２６の出力は１６段のｆ関数を用いた初期ベクトルＩＶの暗号化処理結果であるため、その左８ビットが取り出され、入力①に与えられる平文の最初のブロックM_1 の８ビットとの論理和がＥＸＯＲ回路３０によってとられ、暗号文の最初のブロックC_1 として出力①から出力される。また平文ブロックM_1 の８ビットはレジスタ３３に格納される。

【００６３】同時にレジスタ２８に格納されていた５６ビットが左側に、ｆ関数２６の出力のうち左８ビットが右側に格納される形式で、セレクタ２７を介してレジスタ２１に６４ビットのデータが格納される。

【００６４】図８はＣＦＢモード暗号化動作（その４）の説明図である。ここでは入力の平文ブロックM_2 に対応する最初の処理が実行される。まずレジスタ２１に（その３）で格納された６４ビットのデータが、セレクタ２２，２３を介して２つのＥＸＯＲ回路２４，２５に与えられる。この時セレクタ３１は５６ビットの“０”を、３２はレジスタ３３に格納されているデータ、すなわち入力平文の最初のブロックM_1 のデータを出力しており、ＥＸＯＲ回路２４は６４ビットのうち左側５６ビットをそのままｆ関数２６に出力するが、回路２５は右側８ビットと$M_1$８ビットの排他的論理和を出力することになり、その結果の６４ビットがブロックM_2 の入力に対応する４回のループの処理の最初にｆ関数２６に与えられる。また実質的にこの入力は８ビットシフトされ、レジスタ２８に格納される。そしてｆ関数２６にお

ける４段の変換処理（初期転置を含む）の結果はセレクタ２７を介してレジスタ２１に格納される。

【００６５】図９はＣＦＢモード暗号化動作のタイミングチャートである。動作フローに対する数字は、図５～図８で説明した暗号化処理その１～その４に対応する数字である。平文の最初のブロックM_1の入力に対しては、４つのクロックに対応してその１，その２，その２，その３の処理が実行されて、出力の最初のブロックC_1が得られる。次の入力ブロックM_2以下に対しては、４つのクロックに対応してその４、その２、その２、その３の処理が実行され、出力ブロックC_2以下が得られる。なお、例えば暗号文ブロックC_1は図７で説明したように、４回目のループの最後にｆ関数２６の出力と平文ブロックM_1との排他的論理和の結果として短時間出力されるが、図にはその出力期間を約１クロック分示してある。出力端子❶の前にラッチ回路を設ければこの期間を次の暗号文ブロック出力の前まで延長することも可能である。

【００６６】図１０はＣＦＢモード復号動作（その１）の説明図である。同図においては、暗号文の最初のブロックC_1の入力に対応して、レジスタ２１に格納されている初期ベクトルＩＶの値がセレクタ２２，２３、ＥＸＯＲ回路２４，２５を介してそのままｆ関数２６に与えられる。また初期ベクトルＩＶは８ビット左シフトされるだけでレジスタ２８に格納される。そして初期転置を含むｆ関数による４段の処理の終了後、その結果はセレクタ２７を介してレジスタ２１に格納される。

【００６７】図１１はＣＦＢモード復号動作（その２）の説明図である。同図は、ｆ関数を用いる処理としての４回のループのうちの２回目のループと３回目のループの処理動作の説明図である。レジスタ２１に格納されている６４ビットのデータはセレクタ２２，２３、ＥＸＯＲ回路２４，２５を介してｆ関数２６に与えられ、ｆ関数２６の出力はセレクタ２７を介してレジスタ２１に格納される。

【００６８】図１２はＣＦＢモード復号動作（その３）の説明図である。同図はｆ関数を用いる４回目のループの処理動作を示す。レジスタ２１に格納されている６４ビットのデータは、セレクタ２２，２３、ＥＸＯＲ回路２４，２５を介してｆ関数２６に与えられる。ｆ関数を用いる４段の変換、左右３２ビットの入替え、および逆転置IP^{-1}の処理の後に、その処理結果としての６４ビットのうち左８ビットがＥＸＯＲ回路３０に与えられ、入力❶から入力される最初の暗号文ブロックC_1の８ビットとの排他的論理和がとられ、その結果は出力❶から平文の最初のブロックM_1として出力される。また同時にレジスタ２８に格納されている５６ビットが左側に、入力暗号文ブロックC_1の８ビットが右側に配置される形式で、セレクタ２７を介してレジスタ２１に６４ビットのデータとして格納される。

【００６９】図１３はＣＦＢモード復号動作のタイミングチャートである。動作フローにおける数字は図１０～図１２としてのその１～その３の処理を示す。入力暗号文の最初のブロックC_1の入力に対しては、４つのクロックにおいてその１，その２，その２，その３の処理が実行され、出力平文の最初のブロックM_1が出力される。次の入力暗号文ブロックC_2以下の入力に対しても、全く同様の動作が実行される。

【００７０】図１４はＣＢＣモード暗号化動作（その１）の説明図である。ＣＢＣモードにおいては入力平文ブロックは入力❷から入力され、出力暗号文ブロックは出力❷から出力される。まず入力❷から入力された最初の平文ブロックM_1　６４ビットは、そのうち左５６ビットがセレクタ３１に、右８ビットが３２に与えられる。そしてセレクタ３１はその５６ビットをＥＸＯＲ回路２４に、３２は８ビットを２５に与える。

【００７１】一方レジスタ２１に格納されている初期ベクトルＩＶはセレクタ２２，２３を介してＥＸＯＲ回路２４，２５に与えられ、初期ベクトルＩＶと最初の平文入力ブロックM_1との排他的論理和がとられ、その結果はｆ関数２６に与えられる。そして初期転置ＩＰを含む４段のｆ関数を用いる変換の結果は、セレクタ２７を介してレジスタ２１に格納される。

【００７２】図１５はＣＢＣモード暗号化動作（その２）の説明図である。同図はｆ関数を用いる４回のループのうち、２回目および３回目のループにおける動作の説明図である。レジスタ２１に格納されている６４ビットのデータは、セレクタ２２，２３を介して２つのＥＸＯＲ回路２４，２５に与えられる。この時セレクタ３１，３２はそれぞれ５６ビットの“０”、８ビットの“０”を出力しているため、その６４ビットのデータはそのままｆ関数２６に与えられる。そしてｆ関数を用いる４段の変換処理の結果は、セレクタ２７を介してレジスタ２１に格納される。

【００７３】図１６はＣＢＣモード暗号化動作（その３）の説明図である。同図はｆ関数を用いる４回目のループの動作を説明するものである。レジスタ２１に格納されている６４ビットのデータは、前述と同様にセレクタ２２，２３、ＥＸＯＲ回路２４，２５を介して、そのままｆ関数２６に与えられる。ｆ関数を用いる４段の変換、左右３２ビットの入れ替え、および逆転置IP^{-1}の処理結果は、セレクタ２７を介してレジスタ２１に格納される。レジスタ２１に格納された６４ビットのデータは、ＥＸＯＲ回路３４を介してそのまま出力❷から出力暗号文の最初のブロックC_1として出力さる。この時セレクタ３５は６４ビットの“０”をＥＸＯＲ回路３４に出力している。またレジスタ２１に格納された出力暗号文の最初のブロックC_1は、次の平文ブロックM_2の入力時点で暗号化動作（その１）によって入力ブロックM_2と、前述と同様にして、排他的論理和がとられること

になる。

【００７４】図１７はＣＢＣモード暗号化動作のタイミングチャートである。入力平文ブロックM_1の入力に対応して、４つのクロックにおいてその１、その２、その２、その３の動作が実行され、出力暗号文の最初のブロックC_1が出力される。平文ブロックM_2以下の入力に対する動作は全く同様である。

【００７５】図１８はＣＢＣモード復号動作（その１）の説明図である。ＣＢＣモード復号動作では、初期ベクトルＩＶはレジスタ３７に格納されている。最初の暗号文ブロックC_1が入力②から入力されると、セレクタ３１ではその左側５６ビット、３２では右側８ビットが選択されて、２つのＥＸＯＲ回路２４，２５に与えられる。この時セレクタ２２，２３は共に５６ビットの“０”、８ビットの“０”を出力しており、このため入力暗号文ブロックC_1の６４ビットはそのままｆ関数２６に与えられる。そして初期転置ＩＰを含む、ｆ関数を用いる４段の処理結果は、セレクタ２７を介してレジスタ２１に格納される。

【００７６】図１９はＣＢＣモード復号動作（その２）の説明図である。同図はｆ関数を用いる４回のループのうち２回目、３回目のループにおける動作説明図である。同図においてレジスタ２１に格納されている６４ビットのデータは、セレクタ２２，２３、ＥＸＯＲ回路２４，２５を介してそのままｆ関数２６に与えられ、ｆ関数を用いる４段の変換の処理結果はセレクタ２７を介してレジスタ２１に格納される。またレジスタ３７に格納されている初期ベクトルＩＶは例えば３回目（２回目でもよい）のループの処理時にレジスタ３６に格納される。

【００７７】図２０はＣＢＣモード復号動作（その３）の説明図である。同図において４回目のループとして、レジスタ２１に格納されている６４ビットのデータはそのままｆ関数２６に与えられ、ｆ関数を用いる４段の変換、左右３２ビットの入れ替え、および逆転置IP^{-1}の処理結果がセレクタ２７を介してレジスタ２１に与えられる。レジスタ２１に格納された６４ビットのデータはＥＸＯＲ回路３４に与えられ、レジスタ３６に格納されている初期ベクトルＩＶとの排他的論理和がとられ、出力②から出力平文の最初のブロックM_1として出力される。またこの時次の暗号文ブロックC_2に対する（その３）の動作においてＥＸＯＲ回路３４に出力するためのデータとして、入力暗号文の最初のブロックC_1がレジスタ３７に格納される。

【００７８】図２１はＣＢＣモード復号動作のタイミングチャートである。同図において暗号文の最初のブロックC_1の入力に対しては、４つのクロックにおいてその１、その２、その２、その３の動作が行われ、平文の最初のブロックM_1が出力される。以下暗号文の次のブロックC_2・・・の入力に対する動作は全く同様である。

【００７９】図２２は本発明の第２の実施形態としてのＣＦＢモードとＣＢＣモードの２つの動作を実行する暗号処理回路の構成ブロック図である。図４の第１の実施形態においては、２つのＥＸＯＲ回路２４，２５がＣＢＣモードだけでなく、ＣＦＢモードの暗号化動作においても、ｆ関数２６に与えるべき６４ビットのデータのうちで左側８ビットに対する排他的論理和をとるために用いられていたが、図２２においてはこの排他的論理和がＥＸＯＲ回路４５によってとられる点に基本的な相違がある。このため図４のセレクタ２２，２３は１つのセレクタ４１、ＥＸＯＲ回路２４，２５は１つのＥＸＯＲ回路４２、セレクタ３１，３２は１つのセレクタ４３によって実現されている。

【００８０】図２２の第２の実施形態と図４の第１の実施形態との相違は基本的にＣＦＢモード暗号化動作にあるため、その動作を中心に図２２について説明する。図２２では、前述のＣＦＢモード暗号化動作（その１）に対応して、レジスタ２１に格納されている初期ベクトルＩＶはセレクタ４１、ＥＸＯＲ回路４２を介してそのままｆ関数２６に与えられ、その処理結果はセレクタ４６を介してレジスタ２１に格納される。暗号化動作（その２）における動作も同様である。

【００８１】暗号化動作（その３）に相当して、４回目のループとしてのｆ関数２６の出力のうち左８ビットが、入力①から入力される最初の平文ブロックM_1とＥＸＯＲ回路３０によって排他的論理和がとられ、出力①から最初の暗号文ブロックC_1として出力される。また同時にレジスタ４４に格納されている５６ビットが左側に、ｆ関数２６の出力のうち左８ビットと入力ブロック$M_1$８ビットとの排他的論理和がＥＸＯＲ回路４５によってとられた結果が右側に配置される形式で、セレクタ４６を介して６４ビットのデータがレジスタ２１に格納される。このレジスタ２１に格納されたデータのうち右８ビットはすでに入力との排他的論理和がとられたものであり、次の平文ブロックM_2の入力に対応してセレクタ４１、ＥＸＯＲ回路４２を介してそのままｆ関数２６に与えられることになる。

【００８２】ＣＦＢモード復号、ＣＢＣモード暗号化、およびＣＢＣモード復号の動作は第１の実施形態におけると実質的に同じであるため、その説明を省略する。図２３は、第１の実施形態と第２の実施形態とにおいて、１つのクロックに含まれる処理の説明図である。図４に示した第１の実施形態および図２２に示した第２の実施形態のそれぞれにおいて、ｆ関数２６を用いた１６段の変換は４段ずつ４つのクロックに分けて行われ、１つのブロックデータに対する暗号化処理のために４つのクロックを必要とする点は、いずれの実施形態においても同様である。しかしながら第１の実施形態においては、１つのクロック内で行われる処理の遅延時間の合計が第２の実施形態におけるよりも短くなり、結果的にクロック

の速度を上げることが可能となり、暗号化処理を高速化することができる。

【００８３】図２３のｆ関数による１～４段目の変換を含む１回目のループにおいて、下段の第１の実施形態に対応する図４ではレジスタ２１からの出力による遅延、セレクタ２２，２３による遅延、ＥＸＯＲ回路２４，２５による処理の遅延、４段のｆ関数を用いる変換（初期転置を含む）の遅延、セレクタ２７による遅延に加えて、レジスタ２１におけるデータのセットアップによる遅延が１つのクロック内に含まれることになる。

【００８４】それに対して上段の第２の実施形態に対応する図２２の回路においては、レジスタ２１によるデータ出力の遅延、セレクタ４１による遅延、ＥＸＯＲ回路４２による遅延、ｆ関数を用いる４段の変換による遅延の後に、ＥＸＯＲ回路４５による遅延、セレクタ２７による遅延、レジスタ２１のデータセットアップによる遅延が含まれる。すなわち第１の実施形態に比べてＥＸＯＲ回路４５による遅延が余計に含まれ、第２の実施形態においてはその分だけ第１の実施形態に比べてクロックの周期が長くなり、高速化の面では第１の実施形態より劣ることになる。しかしながらこの点を除けば、第２の実施形態においてもＣＦＢモードとＣＢＣモードとの両方の操作モードを実行する暗号処理回路が実現される。

【００８５】

【発明の効果】以上詳細に説明したように、本発明によればＤＥＳ暗号のＣＢＣモードとＣＦＢモードとの両方の操作モードを実行できる暗号処理回路を実現することが可能となる。また第１の実施形態においては、ＣＦＢモードにおけるモード処理を次のブロックデータに対する処理の最初に移動することによって、ＣＢＣモードの処理を実行する部分と共通の部分を使用することが可能となり、処理ループの最大遅延を短縮することができ、高速処理が可能となる。また同時に排他的論理和の回路の削減も可能となり、ハードウェア量を減少することができ、暗号処理方式の実用性向上に寄与するところが大きい。

【図面の簡単な説明】

【図１】第１の実施形態に対応する原理構成ブロック図である。

【図２】第２の実施形態に対応する原理構成ブロック図である。

【図３】本発明の第１の実施形態において使用されるＣＦＢモード暗号化回路の構成を示すブロック図である。

【図４】本発明の第１の実施形態としての暗号処理回路の構成を示すブロック図である。

【図５】第１の実施形態におけるＣＦＢモード暗号化動作（その１）の説明図である。

【図６】第１の実施形態におけるＣＦＢモード暗号化動作（その２）の説明図である。

【図７】第１の実施形態におけるＣＦＢモード暗号化動作（その３）の説明図である。

【図８】第１の実施形態におけるＣＦＢモード暗号化動作（その４）の説明図である。

【図９】ＣＦＢモード暗号化動作のタイミングチャートである。

【図１０】第１の実施形態におけるＣＦＢモード復号動作（その１）の説明図である。

【図１１】第１の実施形態におけるＣＦＢモード復号動作（その２）の説明図である。

【図１２】第１の実施形態におけるＣＦＢモード復号動作（その３）の説明図である。

【図１３】ＣＦＢモード復号動作のタイミングチャートである。

【図１４】第１の実施形態におけるＣＢＣモード暗号化動作（その１）の説明図である。

【図１５】第１の実施形態におけるＣＢＣモード暗号化動作（その２）の説明図である。

【図１６】第１の実施形態におけるＣＢＣモード暗号化動作（その３）の説明図である。

【図１７】ＣＢＣモード暗号化動作のタイミングチャートである。

【図１８】第１の実施形態におけるＣＢＣモード復号動作（その１）の説明図である。

【図１９】第１の実施形態におけるＣＢＣモード復号動作（その２）の説明図である。

【図２０】第１の実施形態におけるＣＢＣモード復号動作（その３）の説明図である。

【図２１】ＣＢＣモード復号動作のタイミングチャートである。

【図２２】本発明の第２の実施形態としての暗号処理回路の構成を示すブロック図である。

【図２３】第１の実施形態と第２の実施形態における１つのクロック内で行われる処理の比較を示す図である。

【図２４】ＣＢＣモードにおける暗号化処理の基本説明図である。

【図２５】ＣＦＢモードの暗号化処理の基本説明図である。

【図２６】ＣＢＣモードの暗号処理回路の従来例を示す図である。

【図２７】ＣＦＢモードの暗号処理回路の従来例を示す図である。

【符号の説明】

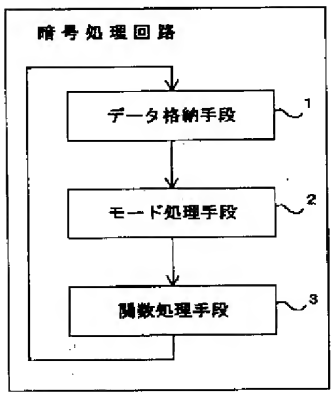
１　データ格納手段
２　関数処理手段
３　モード処理手段
４　第１のモード処理手段
５　第２のモード処理手段
２１，２８，３３，３６，３７，４４　レジスタ
２２，２３，２７，３１，３２，３５，４１，４６　セレクタ

１９　　２０

２４，２５，３０，３４，４２，４５　ＥＸＯＲ回路　　２６　ｆ関数（処理回路）

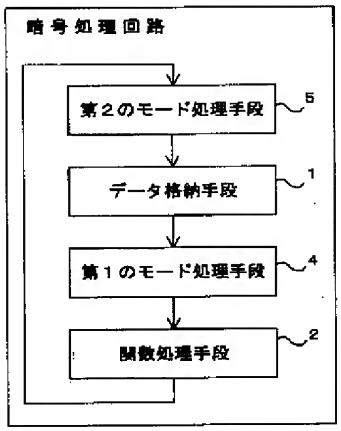
【図１】

第１の実施形態に対応する
原理構成ブロック図



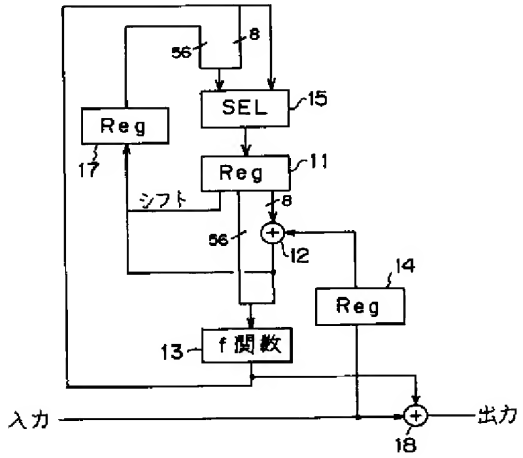
【図２】

第２の実施形態に対応する
原理構成ブロック図



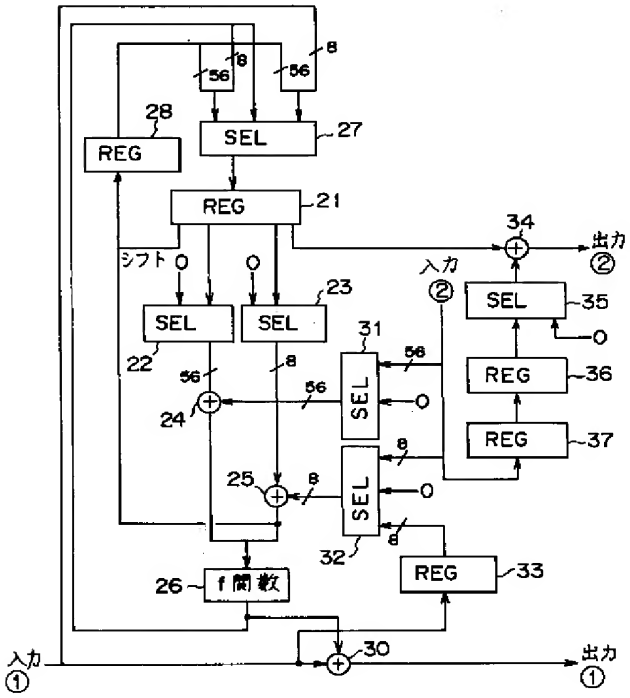
【図３】

本発明の第１の実施形態において使用される
CFBモード暗号化回路の構成を示すブロック図



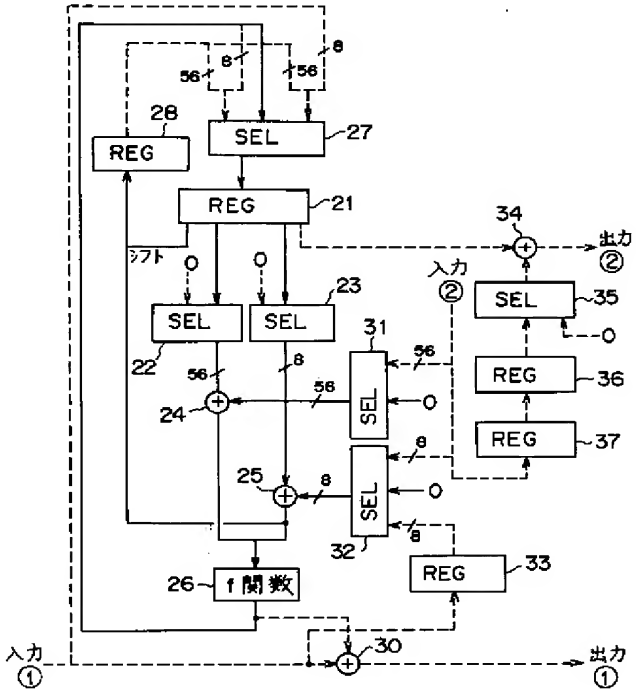
【図４】

本発明の第１の実施形態としての
暗号処理回路の構成を示すブロック図



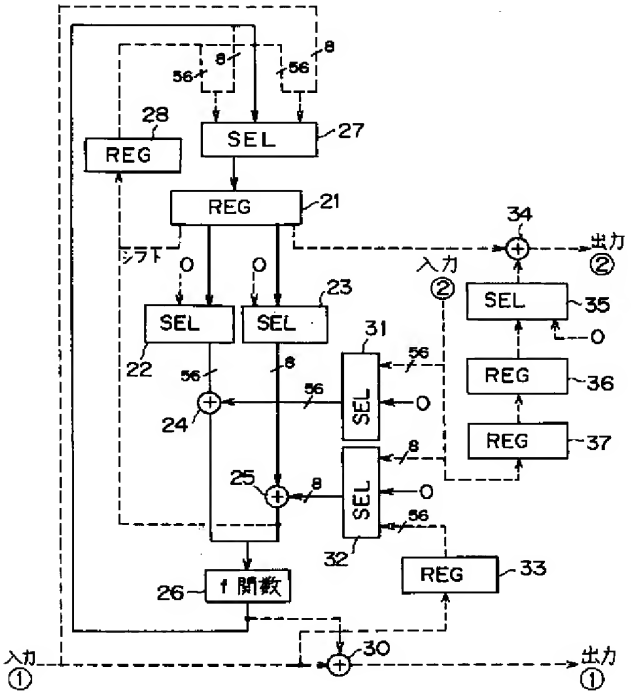
【図５】

第１の実施形態におけるCFBモード暗号化動作
（その1）の説明図



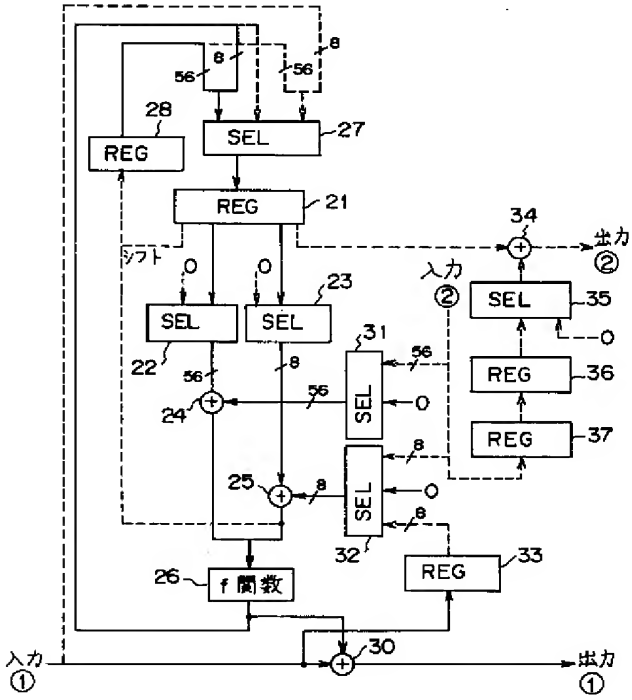
【図６】

第１の実施形態におけるCFBモード化暗号化動作
（その２）の説明図



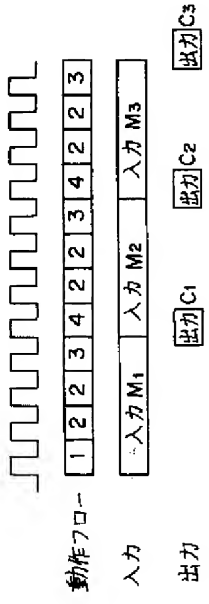
【図７】

第１の実施形態におけるCFBモード暗号化動作
（その３）の説明図



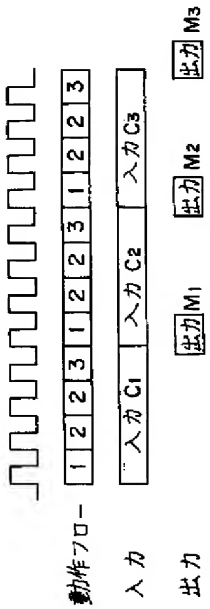
【図９】

CFBモード暗号化動作の
タイミングチャート



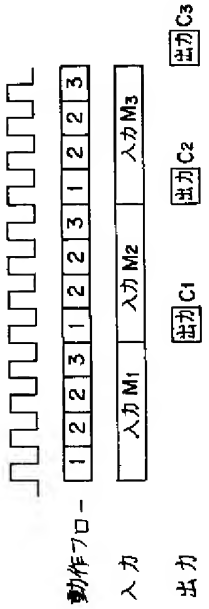
【図１３】

CFBモード復号動作の
タイミングチャート



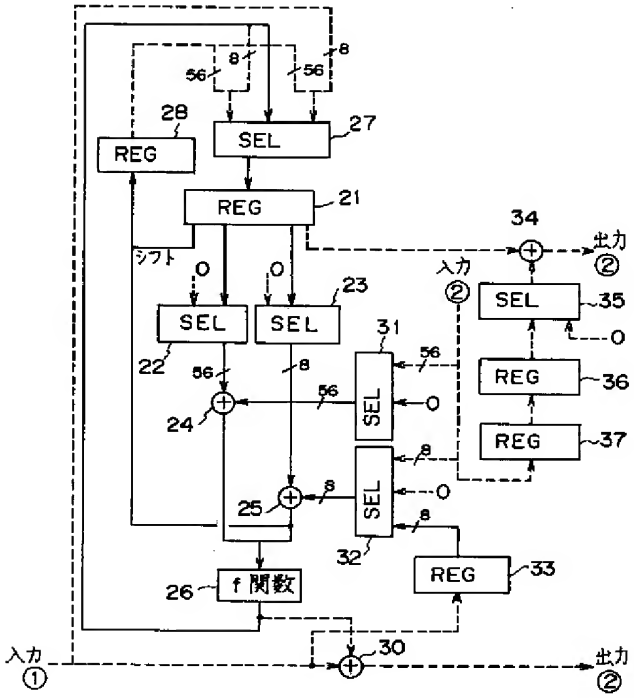
【図１７】

CBCモード暗号化動作の
タイミングチャート



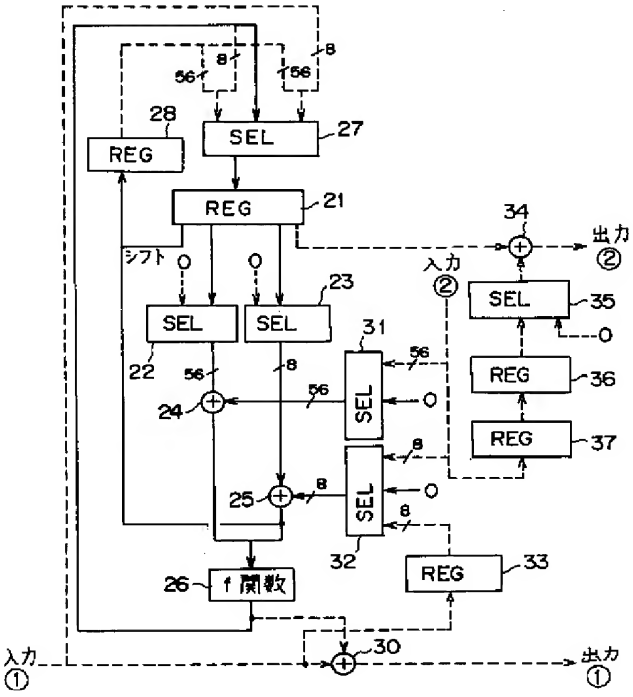
【図８】

第１の実施形態におけるCFBモード暗号化動作（その４）の説明図



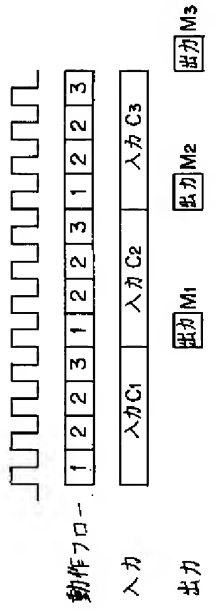
【図１０】

第１の実施形態におけるCFBモード復号動作（その１）の説明図



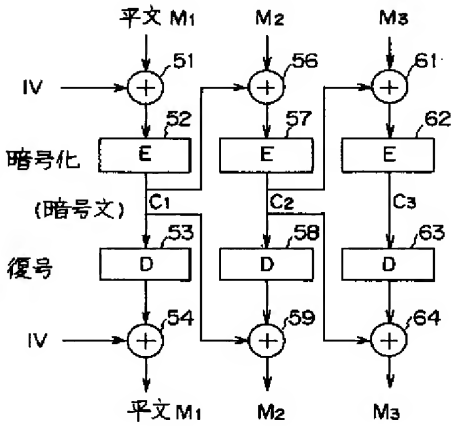
【図２１】

CBCモード復号動作のタイミングチャート



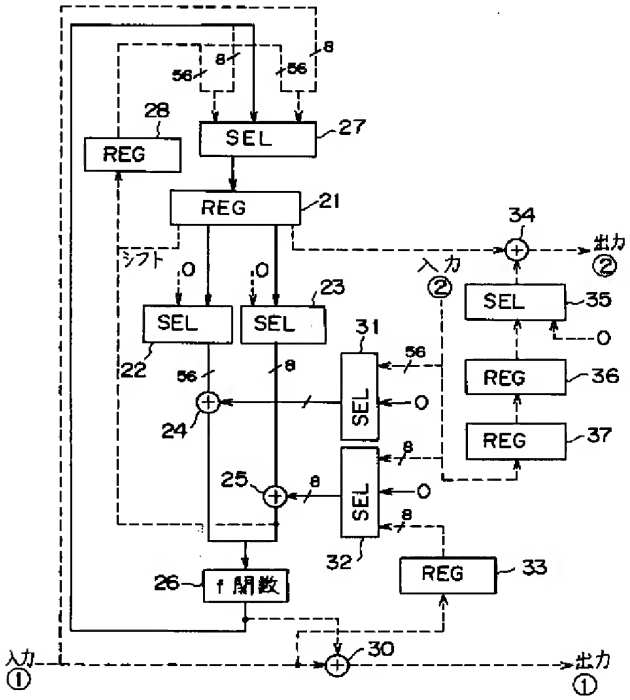
【図２４】

CBCモードにおける暗号化処理の基本説明図



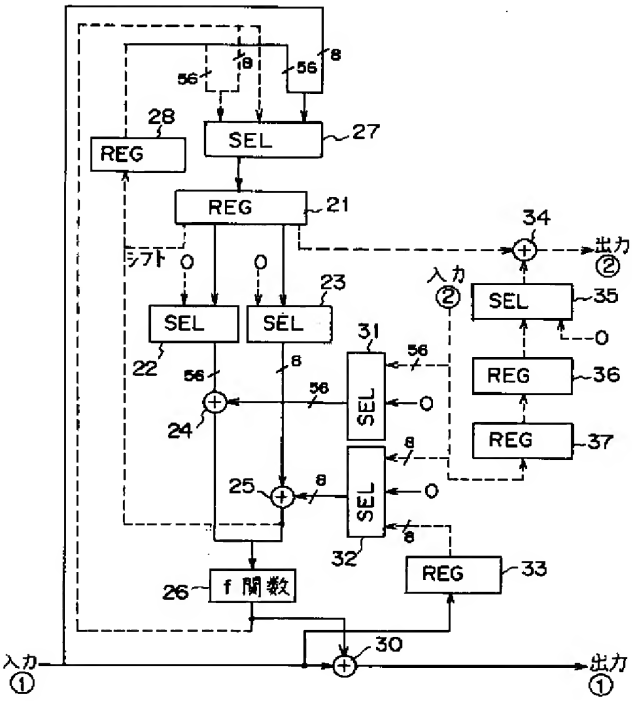
【図１１】

第１の実施形態におけるＣＦＢモード復号動作（その２）の説明図



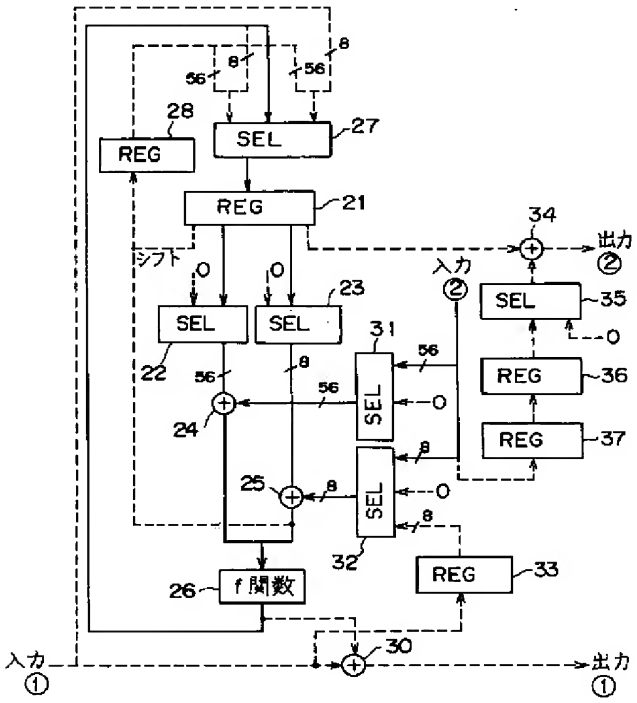
【図１２】

第１の実施形態におけるＣＦＢモード復号動作（その３）の説明図



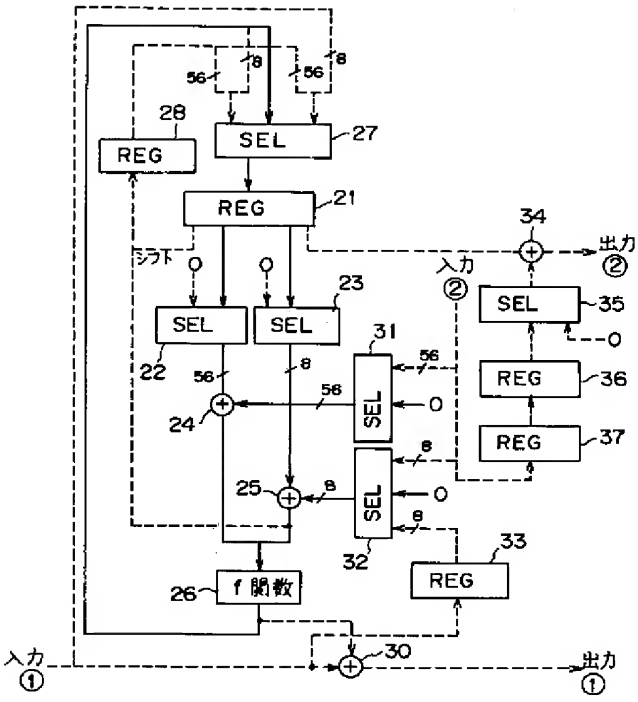
【図１４】

第１の実施形態におけるＣＢＣモード暗号化動作（その１）の説明図



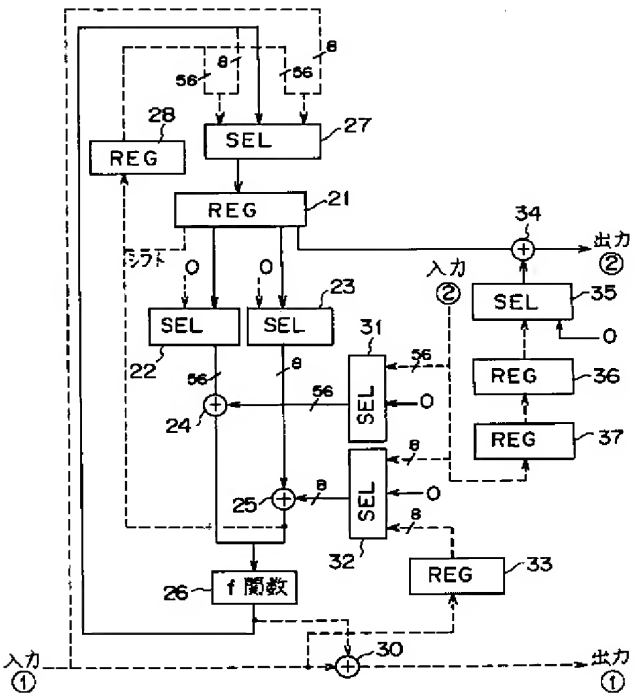
【図１５】

第１の実施形態におけるＣＢＣモード暗号化動作（その２）の説明図



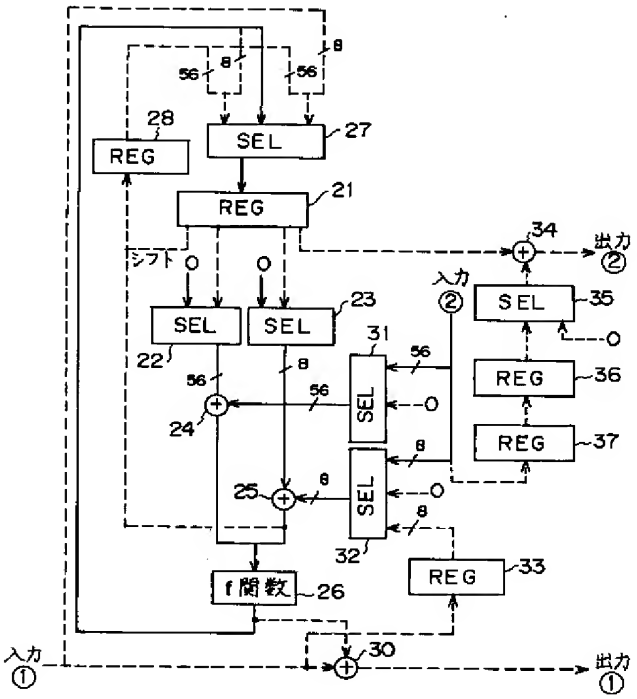
【図１６】

第１の実施形態におけるCBCモード暗号化動作
（その３）の説明図



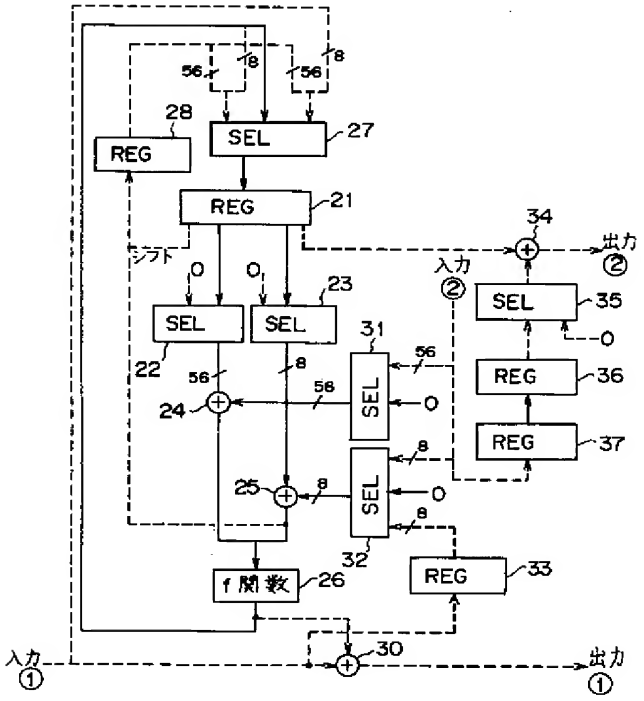
【図１８】

第１の実施形態におけるCBCモード復号動作
（その１）の説明図



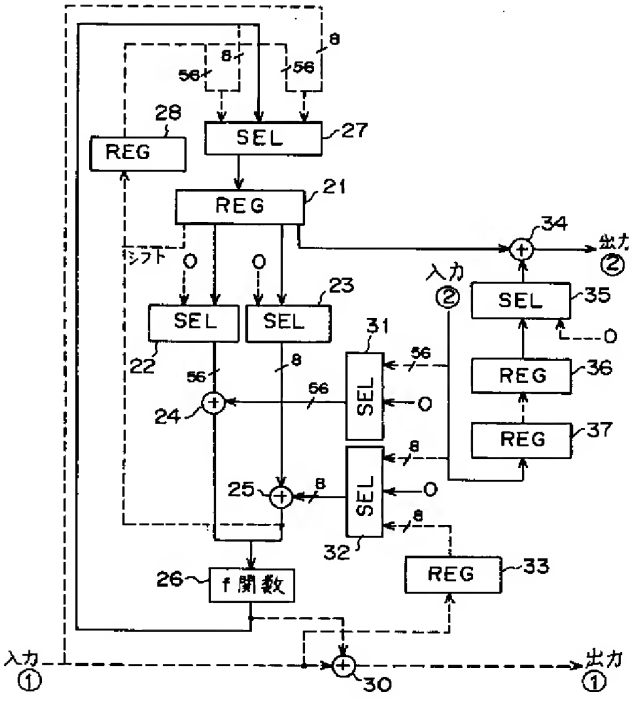
【図１９】

第１の実施形態におけるCBCモード復号動作（その２）の説明図



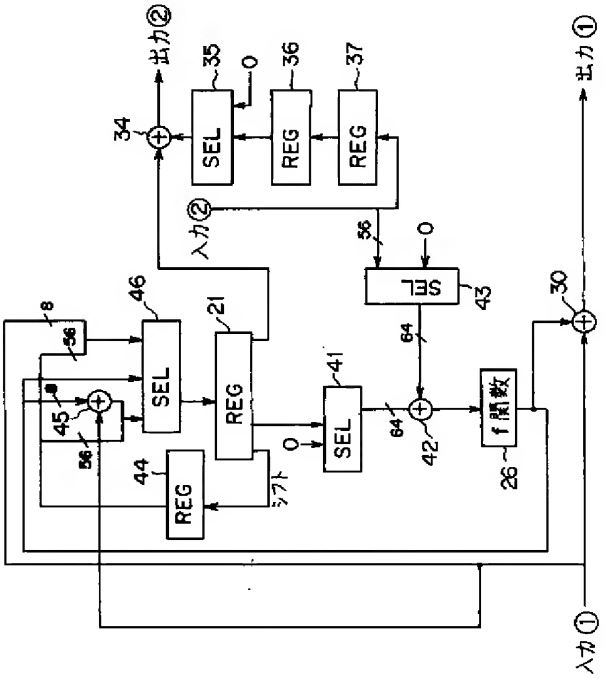
【図２０】

第１の実施形態におけるCBCモード復号動作（その３）の説明図



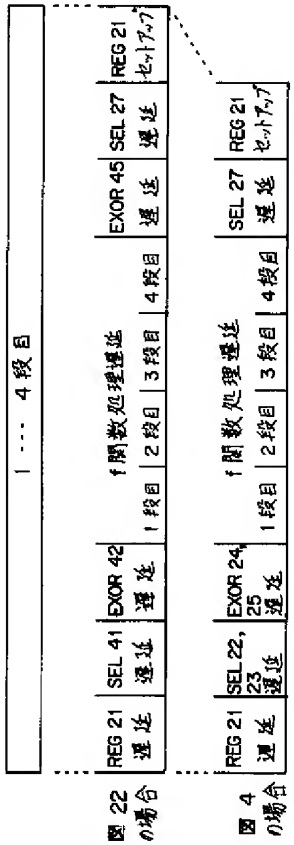
【図２２】

本発明の第２の実施形態としての
暗号処理回路の構成を示すブロック図



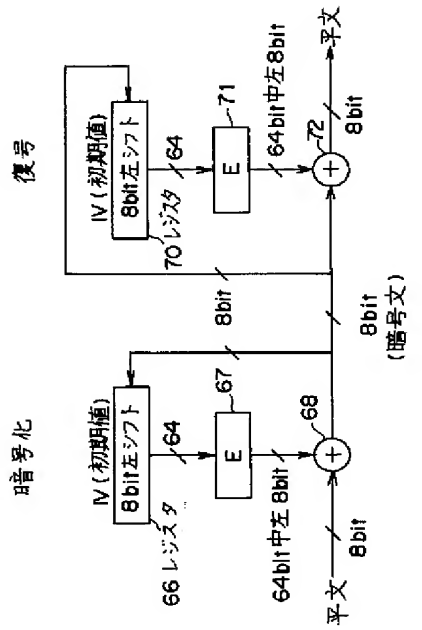
【図２３】

第1の実施形態と第２の実施形態における
1つのクロック内で行われる処理の比較を示す図



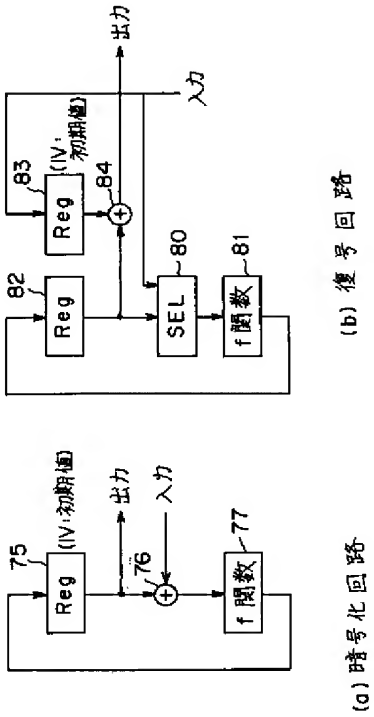
【図25】

CFBモードの暗号化処理の基本説明図



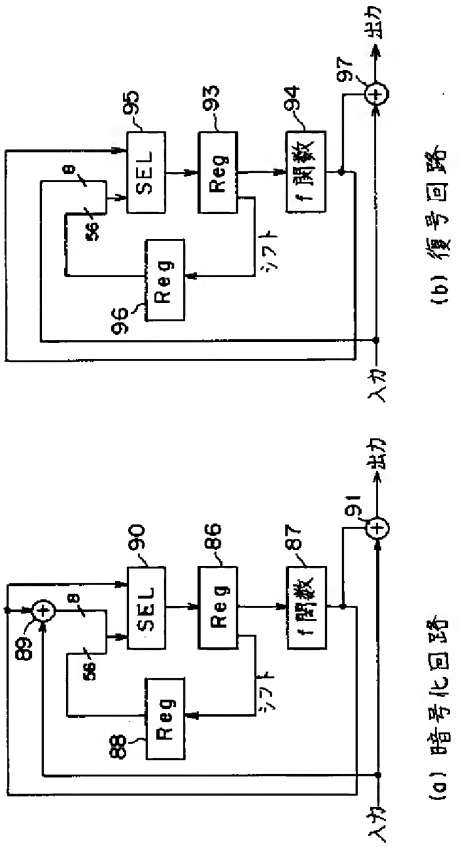
【図26】

CBCモードの暗号処理回路の従来例を示す図



【図２７】

CFBモードの暗号処理回路の従来例を示す図



FULL CONTENTS

[Claim(s)]

[Claim 1] The initial vector used for block encryption processing in the cipher-processing circuit which processes block encryption, a processing mean value, or a data storage means to store the processing final result, The high-speed cipher-processing circuit characterized by having between a function processing means to perform function processing used within block encryption processing, and this data storage means and a function processing means, and having a mode processing means to perform processing corresponding to the operation mode of block encryption.

[Claim 2] The high-speed cipher-processing circuit according to claim 1 characterized by said mode processing means being constituted by the exclusive "or" circuit.

[Claim 3] Claim 1 characterized by for said block cipher being a DES code and said operation mode being CFB mode, or a high-speed cipher-processing circuit given in two.

[Claim 4] The high-speed cipher-processing circuit according to claim 3 characterized by for said function processing means dividing 16 steps of conversion using the nonlinear function as said function processing in encryption of the block data to said DES code into multiple times, and performing it.

[Claim 5] The high-speed cipher-processing circuit according to claim 4 characterized by said function processing means outputting the execution result at each time except the last 1 time of said multiple times to said data storage means.

[Claim 6] The high-speed cipher-processing circuit according to claim 4 characterized by having further an encryption data output means by said function processing means to output the encryption block data to input block data using the processing result of the last round of said multiple times.

[Claim 7] While said mode processing means performs mode processing and outputs this execution result to this function processing means before execution of the time of the

beginning of said multiple times by said function processing means The high-speed cipher-processing circuit according to claim 4 characterized by having further a mode processing result storing means to store a part of execution result of this mode processing.
[Claim 8] The high-speed cipher-processing circuit according to claim 7 characterized by this function processing means outputting this a part of execution result to this data storage means while outputting the data which said mode processing result storing means holds to said data storage means at the time of execution of the time of the last of said multiple times by said function processing means.
[Claim 9] Claim 1 characterized by for said block cipher being a DES code and said operation mode being the CBC mode, or a high-speed cipher-processing circuit given in two.
[Claim 10] The high-speed cipher-processing circuit according to claim 9 characterized by for said function processing means dividing 16 steps of conversion using the nonlinear function as said function processing in encryption of the block data to said DES code into multiple times, and performing it.
[Claim 11] The high-speed cipher-processing circuit according to claim 9 characterized by said function processing means outputting the execution result of each time of said multiple times to said data storage means.
[Claim 12] Before execution of the time of the beginning of said multiple times by said function processing means, mode processing is performed using the block data into which said mode processing means is inputted, and the data stored in said storing means. The high-speed cipher-processing circuit according to claim 9 characterized by outputting the result of this mode processing to this function processing means.
[Claim 13] The initial vector used for block encryption processing in the cipher-processing circuit which processes block encryption, a processing mean value, or a data storage means to store the processing final result, A function processing means to perform function processing used within block encryption processing, The 1st mode processing means which it has between this data storage means and function processing, and performs 1st processing corresponding to the operation mode of block encryption, The high-speed cipher-processing circuit characterized by having between this function processing means and a data storage means, and having the 2nd mode processing means which performs 2nd processing corresponding to the operation mode of block encryption.
[Claim 14] While being at the end time of encryption processing of one block data and outputting a processing result in the cipher-processing method of processing block encryption A processing mean value [in / to the register which stored the initial vector / this encryption processing], or -- storing the processing final result -- this -- one block data -- or -- this -- [the operation corresponding to the operation mode of this block encryption / perform and] between the data of a block of the degree following one block, and the data stored in the

register this result of an operation -- this -- using as a value equivalent to said initial vector in the encryption processing to the following block data following one block -- this -- the cipher-processing method characterized by performing encryption processing to the following block data.

[Claim 15] It is the cipher-processing method according to claim 14 characterized by for said block cipher being a DES code and said operation mode being the CBC mode in CFB mode and ****.

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the high-speed cipher-processing circuit which performs processing in the CBC mode of the DES code as a still more detailed typical block cipher, and CFB mode about the cipher system of digital data.

[0002]

[Description of the Prior Art] Encryption processing of digital data attracts attention in recent years, and the data which is the target of encryption processing is also increasing. For example, the demand of encryption processing of dynamic image data and transaction processing in a server has come out, and the encryption processing itself is asked for improvement in the speed.

[0003] There are a block cipher system which performs 64 bits as a cipher system of digital data, and enciphers for every block, for example as one block, and a stream cipher system which enciphers 1 bit at a time sequentially, for example.

[0004] The target cipher system [this invention] is a block cipher, and is the most typical DES code also in it. DES (data encryption standard) is cryptographic algorithm which is the data code standard defined by Standards Division, the U.S. Department of Commerce, and is used in January, 1977 present most extensively.

[0005] To the DES code, four modes are defined as the operation mode. The mode is four, ECB (electronic code block), CBC (cipher block chaining), CFB (cipher feedback), and OFB (output feed back).

[0006] Since it is two modes, CBC and CFB, it explains that direct relation is in this invention among these four modes about processing of the encryption and the decode in these two operation modes using drawing 24 - drawing 27 .

[0007] Drawing 24 is the basic explanatory view of the encryption and the decode in the CBC mode. Although the plaintext sequence M1 divided into 64 bits, M2, M3, and ... are inputted in this mode first -- M1 as 64 bits of the beginning receive, the exclusive OR 51 with the initial

vector IV is taken, and encryption processing E52 is performed to the result -- the cipher sequence C1, C2, C3, and C1 as 64 bits of the beginning of ... It is obtained. E52 as encryption processing contains the initial transposition IP, 16 steps of conversion of the same structure which used f function, exchange of 32 bits of right and left, and reverse transposition IP-1 here.

[0008] M2 as following 64 bits To an input, it is C1 as first 64 bits of a cipher. The exclusive OR 56 is taken, encryption processing E57 is performed to the result, and it is C2 as 64 bits next to a cipher. ******. 64 bits of ciphers are created at a time like the following.

[0009] In decoding processing, it is C1 as first 64 bits of a cipher. Decoding processing D53 is performed to an input. Although this processing contains the initial transposition IP, 16 steps of conversion using f function, exchange of 32 bits of right and left, and reverse transposition IP-1 like the encryption processing E52, unlike encryption processing, in 16 steps of conversion, it is used in the sequence that a key sequence is reverse. To the result of the decoding processing D53, the exclusive OR 54 with the initial vector IV is taken, and it is M1 as first 64 bits of a plaintext. It is obtained.

[0010] C2 as 64 bits next to a cipher After decoding processing D58 was performed to the input, it is C1 as first 64 bits of a cipher to the result. The exclusive OR 59 is taken and it is M2 as 64 bits next to a plaintext. It is obtained. The plaintext as a decoded result is created like the following.

[0011] Drawing 25 is the basic explanatory view of operation in CFB mode. CFB mode performs random-number-generation operation, a plaintext is divided into every k bits which is 64 or less [1 or more], and the sequence is similarly made into M1, M2, M3, and ... in drawing 24 .

[0012] M1 as a plaintext, for example, 8 bits of the beginning, Corresponding to an input, encryption processing E67 of the initial vector IV stored in the register 66 is performed. This encryption processing itself is the same also in the CBC mode of drawing 24 . only eight of 64 bits as a result of encryption processing of left-hand side are taken out -- M1 the exclusive OR 68 -- it can take -- 8 bit C1 of the beginning of a cipher while being outputted -- 8 bit C1 as the cipher It is stored in the rightmost side of a register 66. That is, it acts as a 8-bit left shift, 8 bits of left-hand side are thrown away, and the initial vector IV stored in the register 66 is C1 most to right-hand side. It will be stored.

[0013] The following 8 bit M2 of a plaintext Corresponding to an input, encryption processing E67 to the contents of storing of a register 66 is performed. 8 bits of lefts of the 64 bits of the result, and M2 the exclusive OR 68 -- it can take -- the following 8 bit C2 of a cipher ****** -- while being outputted, the 8 bits are stored in the rightmost side of a register 66. That is, it acts as a 8-bit left shift, 8 more bits of the initial vector IV are thrown away, and the contents of storing of a register 66 are C2 to the rightmost side of a register 66. It will be stored. The

following 8 bit C3 of a cipher sequence About the following, same operation is completely performed.

[0014] In the decoding processing in CFB mode, it is C1 the first 8 bits of a cipher first. Corresponding to an input, encryption processing E71 of the initial vector IV stored in the register 70 is performed. Unlike the CBC mode of drawing 24 , also in decoding processing, not the processing D but the processing E is used. 8 bits of lefts are taken out among 64 bits as a result of this processing, and it is C1 of a cipher. The exclusive OR 72 is taken and it is M1 the first 8 bits of a plaintext. It is outputted. In encryption processing, it acts as a 8-bit left shift of the contents of the register 70 similarly simultaneously with it, and a cipher is C1 8 bits most to right-hand side. It is stored.

[0015] The following 8 bit C2 of a cipher sequence The processing corresponding to an input is completely the same. That is, encryption processing E71 is performed to the contents of the register 70, and it is 8 bits of lefts in 64 bits of the result, and C2. The exclusive OR 72 is taken and the degree of a plaintext is M2 8 bits. while being obtained As for the contents of the register 70, it acts as a 8-bit left shift, and a cipher is C2 8 bits. It is stored in the rightmost side of a register 70. The cipher sequence C3 and decoding processing of ... are performed like the following.

[0016] Drawing 26 is the conventional parallel of the cipher-processing circuit in the CBC mode. This figure (a) An enciphering circuit is shown. When it can set to drawing 24 in this figure, it is M1 the first 8 bits of a plaintext similarly. If inputted, an exclusive OR with the initial vector IV stored in the register 75 will be taken by the EXOR circuit 76, and the result will be inputted into the f function 77.

[0017] [in order to perform all of 16 steps of conversion using f function as an f function 77, may have 16 steps of conversion circuits but] in order to reduce the amount of hardware, for example 16 steps of conversion is also realizable by having four steps of conversion circuits and carrying out the loop of the intermediate result of processing 4 times through a register 75 and the EXOR circuit 76. In this case, the contents of storing of a register 75 can be outputted to the f function 77 as it is between processings of that loop by setting 64 bits of values of the input to the EXOR circuit 76 to "0" as opposed to all.

[0018] In addition, [since it was easy here, only the f function 77 was shown, but] The initial transposition IP included in the encryption processing E52 of drawing 24 can be considered that it shall be contained in processing of f function of the 1st step, and exchange of 32 bits of the last right and left and reverse transposition IP-1 are contained in the 16th step of processing of f function.

[0019] And 64 bits which it is at the end time of the 4th loop, is outputted from the f function 77, and is stored in a register 75 are the block C1 of the beginning of a cipher. While being outputted by carrying out, it is the block M2 next to an input plaintext. It is M2 at the input time.

It is used in order to take an exclusive OR. Since the following processings are the same as the explanation to drawing 24 , the explanation is omitted.

[0020] Drawing 26 (b) It is the conventional parallel of the CBC decoder circuit. In this figure, it is the block C1 of the beginning of a cipher. An input will give the input to the f function 81 through a selector 80.

[0021] This f function 81 is for performing decoding processing D53 in drawing 24 , and is (a). if it shall have four steps of conversion circuits similarly It is the input cipher block C1 after the end of four operation through a register 82 and a selector 80 of a loop. A decoded result will be outputted to a register 82, and conversion of 32 bits of right and left and reverse transposition IP-1 shall be included in the initial transposition IP and the 16th step of conversion at the 1st-step conversion.

[0022] Block C1 of the beginning of the cipher stored in the register 82 It is inputted into the EXOR circuit 84, an exclusive OR with the initial vector IV stored in the register 83 is taken, and a decoded result is the block M1 of the beginning of a plaintext. It is outputted by carrying out. At this time, it is the next block C2. In order to take a decoded result and an exclusive OR, it is the block C1 of the beginning of an input cipher. It is stored in a register 83. Block C2 next to a cipher Since decoding processing to an input is similarly performed in drawing 24 , the explanation is omitted below.

[0023] Drawing 27 is the conventional parallel of the cipher-processing circuit in CFB mode. This figure (a) An enciphering circuit and (b) A decoder circuit is shown. Drawing 27 (a) It sets and is M1 the first 8 bits of a plaintext. Corresponding to an input, the value of the initial vector IV first stored in the register 86 is outputted to the f function 87.

[0024] The f function 87 performs the encryption processing E67 E52 in drawing 25 , i.e., the encryption processing in drawing 24 , and same processing, and it has four steps of conversion circuits similarly in drawing 26 . 16 steps of conversion by four processings through a selector 90 and a register 86 of a loop ([and] in drawing 26 similarly the first stage) 8 bits of lefts are outputted to the EXOR circuit 91 among 64 bits as a processing result including exchange of 32 bits of right and left, and reverse transposition -- 8 bit M1 of an input plaintext an exclusive OR -- it can take -- block C1 of the beginning of a cipher ****** -- it is outputted.

[0025] As mentioned above, the initial vector IV stored in the register 86 is M1. Although outputted to the f function 87 corresponding to an input, at this time, it is shifted to the 8-bit left, namely, left 8 bits are thrown away, and those contents are stored in a register 88. The number of bits is 56 bits. Only 8 bits of most left-hand side of the output after the 4th loop of the f function 87 are inputted into the EXOR circuit 89. Input M1 An exclusive OR is taken and 8 bits of the result are stored most in right-hand side as 64-bit data through a selector 90 at a register 86 in form that 56 bits stored in the register 88 are arranged on left-hand side. These 64 bits will be outputted to the encryption processing E67 of drawing 25 corresponding to the

input of the next block M2 of a plaintext. M2 Since operation to the following blocks is the same, the following explanation is omitted.
[0026] Drawing 27 (b) It sets and is C1 the first 8 bits of a cipher. at the input time The initial vector IV stored in the register 93 is given to the f function 94. Conversion using 16 steps of f functions etc. is performed by four processings of a loop which mind a selector 95 and a register 93 like the above-mentioned. 8 bits of lefts are outputted to the EXOR circuit 97 among 64 bits as an output of the encryption processing E71 of drawing 25 , and it is the input cipher block C1. An exclusive OR is taken and it is a plaintext M1. It is outputted.
[0027] When the value of the initial vector IV as contents of the register 93 is outputted to the f function 94, simultaneously, it acts as a 8-bit left shift of the contents, and 56 bits is stored in a register 96. And it is at the end time of the 4th loop, and, in left-hand side and 8 bits of input ciphers, the 56 bits are C1. In the form arranged on right-hand side, 64-bit data is stored in a register 93 through a selector 95. This data is given to the f function 94 corresponding to the input of the next cipher block C2. Cipher block C2 Since it is the same about the decoding processing to the following, the explanation is omitted.
[0028]
[Problem to be solved by the invention] In order to accelerate such DES encryption processing, it aims at the improvement in the speed of 16 steps of transform processing which used f function especially conventionally. the conversion circuit of f function -- as much as possible -- a number of stages -- [it connects mostly, and / loop loss is reduced or] The procedure of the conversion containing f function was changed and the method of reducing the delay to 1 time of a loop, or having put in, for example in the CBC encryption processing into the processing cycle of encryption of the mode processing corresponding to the operation mode, i.e., a loop, and reducing the number of clocks required for processing was used. It is indicated by the following United States patent about these methods.
[0029] U.S. the conversion using Patent 5, 381, and 480, Butter et al."System for Translating Encrypted Data", however f function -- as much as possible -- a number of stages -- though it connects mostly The more it increased the number of stages, the more the amount of hardware increased, and since cost became large, there was a problem that a number of stages could also seldom be increased.
[0030] Next, as the operation mode of a DES block cipher, the four operation modes, ECB, CBC, CFB, and OFB, exist as mentioned above. ECB mode only combines DES encryption processing and decoding processing, and the special mode processing circuit for the operation mode is unnecessary. In OFB mode, within the processing loop of DES encryption Processing to the next block can be performed as it is using the processing result of a pre- block, and it is not necessary to take [as special processing corresponding to a mode] generating of delay into consideration to a processing loop that what is necessary is just to take an exclusive OR

between the final disposal result of DES, and input block data.
[0031] On the other hand, it is necessary to take the exclusive OR of the value of the initial vector IV, and input block, for example, and to perform DES processing to this value in the CBC mode. Moreover, it is necessary to perform the left shift of the value of an initial vector at the time of the renewal of contents of the register which stored the initial vector IV in encryption processing, and to set the value of an exclusive OR with a part of processing result of DES processing, and input block to right-hand side in CFB mode.
[0032] That is, the mode processing to CBC is the beginning of DES processing, and there is a difference that mode processing to CFB is performed at the last. These the processings of both are included in the loop of DES processing.
[0033] Although it was desirable to constitute the circuit which can perform all these four operation modes as a processing circuit which processes DES encryption, there was a problem that it was difficult to constitute the processing circuit especially used common to the CBC mode and CFB mode in the former. This originates in the mode processing corresponding to these two modes being [one side / another side] contained at the end at the beginning of the processing loop of DES processing.
[0034] Especially this invention aims at offering the cipher-processing circuit which can perform both the CBC mode and CFB mode, and reducing the processing delay in the circuit as much as possible.
[0035]
[Means for solving problem] Drawing 1 is a principle configuration block figure corresponding to the 1st embodiment of this invention. This figure is a configuration block figure of a high-speed cipher-processing circuit which can perform DES encryption processing in both the CBC mode and CFB mode, for example as the operation mode and which processes block encryption.
[0036] In drawing 1 , the data storage means 1 stores the initial vector used for block encryption processing, a processing mean value, or the processing final result, and is a register.
[0037] The function processing means 2 performs two or more steps of conversion using f function as function processing used within block encryption processing, for example, DES processing. It has the mode processing means 3 between the data storage means 1 and the function processing means 2, and it performs processing corresponding to the operation mode, for example, the CBC mode, or CFB mode of block encryption using the data stored in the data storage means 1.
[0038] In processing in CFB mode, 16 steps of conversion using the nonlinear function as the above-mentioned function processing in the input-block data encryption to a DES code is performed in a step, for example, 4 times, by the function processing means 2. In that case, in

one function processing, four steps of conversion using a nonlinear function is performed.
[0039] The execution result at each time except the last 1 time of processing of these multiple times is stored in the data storage means 1 by the function processing means 2. And since the encryption block data to input block data is outputted using 1 time of this last processing result, it has an encryption data output means further.
[0040] Moreover, while mode processing is performed and that execution result is outputted to the function processing means 2 by the mode processing means 3 before execution of the time of the beginning of processing of these multiple times, it has further a mode processing result storing means by which a part of execution result of mode processing is stored. And while a part of execution result of the above-mentioned mode processing is outputted to the data storage means 1 by the mode processing result storing means at the time of execution of the time of the last of processing of the multiple times by the function processing means 2, a part of execution result is outputted to the data storage means 1 by the function processing means 2.
[0041] In the time of execution of encryption processing in the CBC mode, in CFB mode, similarly, 16 steps of conversion using the nonlinear function as function processing in encryption of the block data to the above-mentioned DES code divides into multiple times, and is performed. And the execution result of each time of processing of these multiple times is stored in the data storage means 1 by the function processing means 2.
[0042] Moreover, before execution of the time of the beginning of processing of the multiple times by the function processing means 2, mode processing using the block data inputted by the mode processing means 3 and the data stored in the data storage means is performed, and the result of the mode processing is outputted to the function processing means 2.
[0043] Drawing 2 is a principle configuration block figure corresponding to the 2nd embodiment of this invention. In addition to the 1st mode processing means 4 corresponding to the mode processing means 3 of drawing 1 , in this figure, it has between the function processing means 2 and the data storage means 1, and the 2nd mode processing means 5 which performs 2nd processing corresponding to the operation mode of block encryption is added.
[0044] In the 2nd embodiment of this invention although the cipher-processing circuit can perform the operation mode in both CFB mode and the CBC mode similarly in the 1st embodiment Fundamental mode processing in CFB mode is performed by the 2nd mode processing means 5 to the 1st mode processing means 4 performing the fundamental portion of the mode processing in the CBC mode.
[0045] [conversion] in CFB mode although it divides into multiple times similarly and 16 steps of conversion using the nonlinear function by the function processing means 2 is performed also in the 1st embodiment The data stored in the data storage means 1 in the first time is given to the function processing means 2 as it is, without performing special processing by the

1st mode processing means 3. The processing result at each time except the time of the last of processing of the multiple times by the function processing means 2 is stored in the data storage means 1.

[0046] While the encryption block data to input block data is similarly outputted in the 1st embodiment using the processing result of the last round by the function processing means 2 By the 2nd mode processing means, it is performed by the 2nd mode processing using the part and input data block of the processing result, and The result, Before execution of the time of the beginning of multiple times, some data stored in the data storage means 1 is stored in the data storage means 1, and it is used for encryption processing of the following block data.

[0047] In the CBC mode, according to the mode processing means 3 against the 1st embodiment, same processing is performed by the 1st mode processing means 4, and also in the 1st embodiment, same processing is performed as a result, without the 2nd mode processing means performing any operation substantially.

[0048] While being at the end time of encryption processing of one block data and outputting a processing result in the cipher-processing method of processing DES block encryption, for example, in the cipher-processing method of this invention The processing mean value in the encryption processing or the processing final result is stored in the register in which the initial vector was stored.

[0049] And the operation corresponding to the operation mode of the block encryption is performed between the data stored in the register, its one block data, or the following block data following the one block. Furthermore, the result of an operation is used as a value equivalent to the initial vector in the encryption processing to the following block data following the one block, and encryption processing to the following block data is performed.

[0050] In the encryption processing method of this invention, the initial vector is first stored in the above-mentioned register. After the operation corresponding to the operation mode is first performed between input block data in the CBC mode as opposed to the initial vector In CFB mode, without performing substantially, corresponding to the result of an operation, 16 steps of conversion using a nonlinear function divides the special operation corresponding to the operation mode into multiple times, and it is performed. The method of this encryption processing is used in the 1st above-mentioned embodiment.

[0051] As explained above, according to this invention, the cipher-processing circuit which performs both the CBC mode and CFB mode is realizable. Moreover, it becomes possible to communalize the mode processing means corresponding to two modes in the 1st embodiment.

[0052]

[Mode for carrying out the invention] Drawing 3 shows the composition of the enciphering circuit in the CFB mode used in the 1st embodiment of this invention. This figure is drawing 27

(a). It is drawing 27 (a) in order to use it combining the enciphering circuit in the CBC mode, although it corresponds to the shown conventional parallel. The EXOR circuit 89 set and used is omitted, instead the EXOR circuit 12 is used for the output side of the register 11.
[0053] The initial vector IV is stored in the register 11 in drawing 3 . 8 bit M1 of the beginning of a plaintext At the input time, the value of the initial vector IV is given to the f function (processing circuit) 13 from a register 11. At this time, the value of the initial vector IV stored in the register 11 is given to the f function 13 from a register 14 as it is by outputting "0", for example to the EXOR circuit 12.
[0054] The f function 13 is drawing 27 (a). If it can set, it will consist of a conversion circuit using four steps of f functions similarly, and 16 steps of conversion using f function shall be performed by four processings through a selector 15 and a register 11 of a loop. Moreover, the initial transposition IP, exchange of 32 bits of right and left to the result of the 16th step, and reverse transposition IP-1 shall be included as mentioned above.
[0055] 8 bits of lefts are given to the EXOR circuit 18 among the outputs of the f function 13 after 4 times of loops, and it is the input plaintext block M1. An exclusive OR is taken and it is the first cipher block C1. It is outputted by carrying out.
[0056] Block M1 [56 bits of left-hand side] among the contents although 8 bits of right-hand side is outputted to the register 17 side from a register 11 as it is in the form given through the EXOR circuit 12 when the initial vector IV is outputted from a register 11 corresponding to an input The 64 bits are stored in a register 17, after 8 bit shifts are carried out and 8 bits of most left-hand side are thrown away. And 64-bit data is stored in a register 11 through a selector 15 in form that are at the end time of 4 times of loops, and 56 bits of the contents are stored in left-hand side, and 8 bits of lefts of the outputs of the f function 13 are stored in right-hand side. Moreover, input-block M1 8 bit is stored in a register 14 at this time. 64 bits stored in the register 11 here are the following input block M2, if it remains as it is. It cannot use as an object of DES encryption processing to the f function 13 at the input time.
[0057] Block M2 At the input time, 8 bits of right-hand side is outputted to the EXOR circuit 12 among the contents of storing of a register 11. An exclusive OR with 8 bits of the first block M1 stored in the register 14 is taken, and it is inputted into the f function 13 with 56 bits of left-hand side outputted as it is from a register 11. 8 bit shifts are carried out to the left like the above-mentioned, and 56 bits of 64 bits [which becomes an input to the f function 13 substantially at this time] are stored in a register 17. This is the next block M3. It is for the data origination to the f function 13 at the input time.
[0058] It is at the end time of 4 times of the loops through the f function 13, a selector 15, and a register 11, the exclusive OR of 8 bits of lefts and input M2 8 bit is taken by the EXOR circuit 18 among the outputs of the f function 13, and it is the block C2 next to a cipher sequence. It is outputted by carrying out. The contents of storing of a register 14 shall not be outputted

between this four loop processing, but "0" shall be given to the EXOR circuit 12. Since the same processing as the following is performed, the explanation is omitted.

[0059] Drawing 4 is the configuration block figure of the cipher-processing circuit as the 1st embodiment of this invention. This figure is a configuration block figure of the cipher-processing circuit which performs processing in the mode in both CFB mode and the CBC mode. CFB mode encryption operation using this circuit is explained using drawing 5 - drawing 9 .

[0060] Drawing 5 is the explanatory view of CFB mode encryption operation (the 1). The dashed line has shown the portion which is a solid line and is not used about the connection portion of the circuit used here. It is the first plaintext block M1 first. The initial vector IV stored in the register 21 corresponding to the input (from **) gives the f function 26 through selectors 22 and 23 and the EXOR circuits 24 and 25, and it is ****. At this time, selectors 22 and 23 choose the output from a register 21, and the data (56 bits and 8 bits) of "0" is inputted through selectors 31 and 32 to the EXOR circuits 24 and 25. For this reason, the value of the initial vector IV stored in the register 21 is given to the f function 26 as it is. Moreover, 8 bit shifts of the value of IV stored in the register 21 are only carried out, and it is substantially stored in a register 28 as it is. And the processing result of four steps of conversion circuits contained in the f function 26 is stored in a register 21 through a selector 27.

[0061] Drawing 6 is the explanatory view of CFB mode encryption operation (the 2). This figure is an explanatory view of processing operation in the 2nd time and the 3rd loop among 4 times of the loops through the f function 26, a selector 27, and a register 21. Since selectors 22 and 23 choose the output from a register 21 in these loops and selectors 31 and 32 are outputting "0" and "0" to the EXOR circuits 24 and 25, respectively, [56 bits] [8-bit] The data of a register 21 is given to the f function 26 as it is, and four steps of processing results are stored in a register 21 through a selector 27.

[0062] Drawing 7 is the explanatory view of CFB mode encryption operation (the 3). In this figure, the processing result of the 3rd loop stored in the register 21 is given to the f function 26 as it is through selectors 22 and 23 and the EXOR circuits 24 and 25 like the above-mentioned. Since the output of the f function 26 is the encryption processing result of the initial vector IV which used 16 steps of f functions, Block M1 of the beginning of the plaintext which 8 bits of the left is taken out and given to input ** An OR with 8 bits is taken by the EXOR circuit 30, and it is the block C1 of the beginning of a cipher. It carries out and is outputted from output **. Moreover, plaintext block M1 8 bits is stored in a register 33.

[0063] 64-bit data is stored in a register 21 through a selector 27 in form that 56 bits simultaneously stored in the register 28 are stored in left-hand side, and 8 bits of lefts are stored in right-hand side among the outputs of the f function 26.

[0064] Drawing 8 is the explanatory view of CFB mode encryption operation (the 4). Plaintext

block M2 of an input [here] The first processing in which it corresponds is performed. The 64-bit data first stored in the register 21 by (the 3) is given to two EXOR circuits 24 and 25 through selectors 22 and 23. At this time, a selector 31 is the block M1 of the beginning of the data with which 32 is stored in the register 33 in "0", i.e., an input plaintext. although data is outputted and the EXOR circuit 24 outputs 56 bits of left-hand side to the f function 26 as it is among 64 bits [56-bit] A circuit 25 will output 8 bits of right-hand side, and an M18 bit exclusive OR, and 64 bits of the result are block M2. It is given to the f function 26 at the beginning of four processings of a loop corresponding to an input. Moreover, substantially, 8 bit shifts of this input are carried out, and it is stored in a register 28. And the result of four steps of transform processing (initial transposition is included) in the f function 26 is stored in a register 21 through a selector 27.

[0065] Drawing 9 is the timing chart of CFB mode encryption operation. encryption **** which explained the number over a flow of operation by drawing 5 - drawing 8 -- the 1- it is a number corresponding to the 4. Block M1 of the beginning of a plaintext To an input, corresponding to four clocks, processing of the 1, its 2, and the 2 and its 3 is performed, and it is the block C1 of the beginning of an output. It is obtained. The following input block M2 To the following, corresponding to four clocks, processing of the 4, its 2, and the 2 and its 3 is performed, and it is the output block C2. The following is obtained. In addition, cipher block C1 As drawing 7 explained, they are the output of the f function 26, and the plaintext block M1 to the last of the 4th loop. Although a short-time output is carried out as a result of an exclusive OR, the output period is shown in the figure by about 1 clock. If a latch circuit is prepared before output terminal **, it is also possible to extend this period before the next cipher block output.

[0066] Drawing 10 is the explanatory view of CFB mode decode operation (the 1). In this figure, it is the block C1 of the beginning of a cipher. The value of the initial vector IV stored in the register 21 corresponding to the input is given to the f function 26 as it is through selectors 22 and 23 and the EXOR circuits 24 and 25. Moreover, the initial vector IV is stored in a register 28 only by acting as a 8-bit left shift. And the result is stored in a register 21 through a selector 27 after the end of four steps of processings by f function including initial transposition.

[0067] Drawing 11 is the explanatory view of CFB mode decode operation (the 2). This figure is an explanatory view of processing operation of the 2nd loop of 4 times of the loops as processing which use f function, and the 3rd loop. The 64-bit data stored in the register 21 is given to the f function 26 through selectors 22 and 23 and the EXOR circuits 24 and 25, and the output of the f function 26 is stored in a register 21 through a selector 27.

[0068] Drawing 12 is the explanatory view of CFB mode decode operation (the 3). This figure shows processing operation of the 4th loop which uses f function. The 64-bit data stored in the register 21 is given to the f function 26 through selectors 22 and 23 and the EXOR circuits 24

and 25. Four steps of conversion and the exchange of 32 bits of right and left, and reverse transposition IP-1 after processing [using f function] The first cipher block C1 which 8 bits of lefts are given to the EXOR circuit 30 among 64 bits as the processing result, and is inputted from input ** An exclusive OR with 8 bits is taken and the result is the block M1 of the beginning of a plaintext from output **. It is outputted by carrying out. Moreover, to left-hand side, 56 bits simultaneously stored in the register 28 are the input cipher block C1. In the form arranged on right-hand side, 8 bits is stored in a register 21 as 64-bit data through a selector 27.

[0069] Drawing 13 is the timing chart of CFB mode decode operation. the number in a flow of operation -- the 1- as drawing 10 - drawing 12 -- the processing of 3 is shown. Block C1 of the beginning of an input cipher To an input, in four clocks, processing of the 1, its 2, and the 2 and its 3 is performed, and it is the block M1 of the beginning of an output plaintext. It is outputted. The next input cipher block C2 Same operation is completely performed also to the following inputs.

[0070] Drawing 14 is the explanatory view of the CBC mode encryption operation (the 1). In the CBC mode, an input plaintext block is inputted from input **, and an output cipher block is outputted from output **. Before long, 56 bits of lefts are given to a selector 31, and, as for the plaintext block M1 64 bit of the beginning first inputted from input **, 8 bits of rights are given to 32. And as for a selector 31, the 56 bits are given to the EXOR circuit 24, and 32 gives 8 bits to 25.

[0071] The EXOR circuits 24 and 25 are given through selectors 22 and 23, and the initial vector IV stored in the register 21 on the other hand is the plaintext input block M1 of the initial vector IV and the beginning. An exclusive OR is taken and the result is given to the f function 26. And the result of the conversion using four steps of f functions including the initial transposition IP is stored in a register 21 through a selector 27.

[0072] Drawing 15 is the explanatory view of the CBC mode encryption operation (the 2). This figure is an explanatory view of operation in the 2nd time and the 3rd loop among 4 times of the loops which use f function. The 64-bit data stored in the register 21 is given to two EXOR circuits 24 and 25 through selectors 22 and 23. Since selectors 31 and 32 are outputting "0" and "0", respectively at this time, that 64-bit data is given to the f function 26 as it is. [56 bits] [8-bit] And the result of four steps of transform processing using f function is stored in a register 21 through a selector 27.

[0073] Drawing 16 is the explanatory view of the CBC mode encryption operation (the 3). This figure illustrates operation of the 4th loop which uses f function. The 64-bit data stored in the register 21 is given to the f function 26 as it is through selectors 22 and 23 and the EXOR circuits 24 and 25 like the above-mentioned. Four steps of conversion and the exchange of 32 bits of right and left using f function, and the processing result of reverse transposition IP-1 are

stored in a register 21 through a selector 27. the 64-bit data stored in the register 21 minds the EXOR circuit 34 -- as it is -- block C1 of the beginning of output ** to an output cipher ****** -- **** -- last ** At this time, the selector 35 is outputting "0" to the EXOR circuit 34. [64-bit] Moreover, block C1 of the beginning of the output cipher stored in the register 21 The next plaintext block M2 It is the input block M2 by encryption operation (the 1) at the input time. An exclusive OR will be taken like the above-mentioned.

[0074] Drawing 17 is the timing chart of the CBC mode encryption operation. Input plaintext block M1 Corresponding to an input, in four clocks, operation of the 1, its 2, and the 2 and its 3 is performed, and it is the block C1 of the beginning of an output cipher. It is outputted. Plaintext block M2 Operation to the following inputs is completely the same.

[0075] Drawing 18 is the explanatory view of the CBC mode decode operation (the 1). The initial vector IV is stored in the register 37 in the CBC mode decode operation. The first cipher block C1 If inputted from input **, at a selector 31, by 56 bits of the left-hand side, and 32, 8 bits of right-hand side will be chosen, and two EXOR circuits 24 and 25 will be given. At this time, both the selectors 22 and 23 are outputting "0" and "0", and, for this reason, are the input cipher blocks C1. 64 bits is given to the f function 26 as they are. [56 bits] [8-bit] And four steps of processing results using f function including the initial transposition IP are stored in a register 21 through a selector 27.

[0076] Drawing 19 is the explanatory view of the CBC mode decode operation (the 2). This figure is an explanatory view of operation in the 2nd time and the 3rd loop among 4 times of the loops which use f function. The 64-bit data stored in the register 21 in this figure is given to the f function 26 as it is through selectors 22 and 23 and the EXOR circuits 24 and 25, and the processing result of four steps of conversion using f function is stored in a register 21 through a selector 27. Moreover, the initial vector IV stored in the register 37 is stored in a register 36 at the time of processing of the 3rd loop (the 2nd time is sufficient).

[0077] Drawing 20 is the explanatory view of the CBC mode decode operation (the 3). The 64-bit data stored in the register 21 as 4th loop in this figure is given to the f function 26 as it is, and four steps of conversion and the exchange of 32 bits of right and left using f function, and the processing result of reverse transposition IP-1 are given to a register 21 through a selector 27. The EXOR circuit 34 is given, an exclusive OR with the initial vector IV stored in the register 36 is taken, and the 64-bit data stored in the register 21 is the block M1 of the beginning of an output plaintext from output **. It is outputted by carrying out. Moreover, it is the next cipher block C2 at this time. As data for outputting to the EXOR circuit 34 in operation receiving (that 3), it is the block C1 of the beginning of an input cipher. It is stored in a register 37.

[0078] Drawing 21 is the timing chart of the CBC mode decode operation. In this figure, it is the block C1 of the beginning of a cipher. To an input, in four clocks, operation of the 1, its 2, and

the 2 and its 3 is performed, and it is the block M1 of the beginning of a plaintext. It is outputted. Operation to the input of next block C2 ... of a cipher is completely the same below.
[0079] Drawing 22 is the configuration block figure of the cipher-processing circuit which performs two operation, the CFB mode as the 2nd embodiment of this invention, and the CBC mode. [in order that two EXOR circuits 24 and 25 might take the exclusive OR to 8 bits of left-hand side also in encryption operation in not only the CBC mode but CFB mode in the 1st embodiment of drawing 4 among the 64-bit data which should be given to the f function 26, were used, but] In drawing 22 , there is a difference with this exclusive OR fundamental to the point taken by the EXOR circuit 45. For this reason, in the EXOR circuits 24 and 25, as for the selectors 22 and 23 of drawing 4 , one selector 41 is realized, and one EXOR circuit 42 and selectors 31 and 32 are realized by one selector 43.
[0080] Since CFB mode encryption operation has fundamentally the difference with the 2nd embodiment of drawing 22 , and the 1st embodiment of drawing 4 , it explains drawing 22 focusing on the operation. In drawing 22 , the initial vector IV stored in the register 21 corresponding to the above-mentioned CFB mode encryption operation (the 1) is given to the f function 26 as it is through a selector 41 and the EXOR circuit 42, and the processing result is stored in a register 21 through a selector 46. Operation in encryption operation (the 2) is also the same.
[0081] The first plaintext block M1 into which it is equivalent to encryption operation (the 3), and 8 bits of lefts are inputted from input ** among the outputs of the f function 26 as 4th loop An exclusive OR is taken by the EXOR circuit 30 and it is outputted as the first cipher block C1 from output **. [56 bits simultaneously stored in the register 44 / left-hand side / moreover, form that the result by which the exclusive OR of 8 bits of lefts and input-block M1 8 bit was taken by the EXOR circuit 45 is arranged at right-hand side] among the outputs of the f function 26 64-bit data is stored in a register 21 through a selector 46. An exclusive OR with an input is already taken among the data stored in this register 21, and 8 bits of rights are the next plaintext blocks M2. Corresponding to an input, it will be given to the f function 26 as it is through a selector 41 and the EXOR circuit 42.
[0082] Also in the 1st embodiment, since it is substantially the same, operation of CFB mode decode, the CBC mode encryption, and the CBC mode decode omits the explanation. Drawing 23 is the explanatory view of the processing included in one clock in the 1st embodiment and 2nd embodiment. In each of the 2nd embodiment shown in the 1st embodiment and drawing 22 which were shown in drawing 4 The point which 16 steps of conversion using the f function 26 is performed by dividing into every four-step four clocks, and needs four clocks for the encryption processing to one block data is the same also in which embodiment. However, in the 1st embodiment, it becomes short, it becomes possible to gather the speed of a clock as a result, and encryption processing can be accelerated rather than the sum total of the time

delay of the processing performed within one clock can set to the 2nd embodiment.
[0083] In the 1st loop including the 1-4th step of conversion by f function of drawing 23 Delay according to the output from a register 21 at drawing 4 corresponding to the 1st embodiment of the lower berth, In addition to the delay of conversion (initial transposition is included) using delay by selectors 22 and 23, delay of processing by the EXOR circuits 24 and 25, and four steps of f functions, and delay by a selector 27, delay by the setup of the data in a register 21 will be included in one clock.
[0084] In the circuit of drawing 22 corresponding to the 2nd embodiment of an upper case to it Delay by the EXOR circuit 45, delay by a selector 27, and delay by the data setup of a register 21 are included after delay of the data output by a register 21, delay by a selector 41, delay by the EXOR circuit 42, and delay by four steps of conversion using f function. That is, compared with the 1st embodiment, delay by the EXOR circuit 45 is included too many, in the 2nd embodiment, compared with the 1st embodiment, only the part becomes long and the cycle of a clock will be inferior to the 1st embodiment in it in respect of improvement in the speed. However, if this point is removed, the cipher-processing circuit which performs the operation mode in both CFB mode and the CBC mode also in the 2nd embodiment will be realized.
[0085]
[Effect of the Invention] As explained to details above, according to this invention, it becomes possible to realize the cipher-processing circuit which can perform the operation mode in both the CBC mode of a DES code, and CFB mode. Moreover, by moving the mode processing in CFB mode to the beginning of processing to the following block data in the 1st embodiment It becomes possible to use the portion which performs processing in the CBC mode, and a common portion, and the maximum delay of a processing loop can be shortened and high speed processing becomes possible. Moreover, the place which the reduction of the circuits of an exclusive OR of is also attained simultaneously, can decrease in number the amount of hardware, and contributes to the improvement in practicality of a cipher-processing method is large.

[Brief Description of the Drawings]
[Drawing 1] It is a principle configuration block figure corresponding to the 1st embodiment.
[Drawing 2] It is a principle configuration block figure corresponding to the 2nd embodiment.
[Drawing 3] It is the block diagram showing the composition of the CFB mode enciphering circuit used in the 1st embodiment of this invention.
[Drawing 4] It is the block diagram showing the composition of the cipher-processing circuit as the 1st embodiment of this invention.

[Drawing 5] It is the explanatory view of CFB mode encryption operation (the 1) in the 1st embodiment.
[Drawing 6] It is the explanatory view of CFB mode encryption operation (the 2) in the 1st embodiment.
[Drawing 7] It is the explanatory view of CFB mode encryption operation (the 3) in the 1st embodiment.
[Drawing 8] It is the explanatory view of CFB mode encryption operation (the 4) in the 1st embodiment.
[Drawing 9] It is the timing chart of CFB mode encryption operation.
[Drawing 10] It is the explanatory view of CFB mode decode operation (the 1) in the 1st embodiment.
[Drawing 11] It is the explanatory view of CFB mode decode operation (the 2) in the 1st embodiment.
[Drawing 12] It is the explanatory view of CFB mode decode operation (the 3) in the 1st embodiment.
[Drawing 13] It is the timing chart of CFB mode decode operation.
[Drawing 14] It is the explanatory view of the CBC mode encryption operation (the 1) in the 1st embodiment.
[Drawing 15] It is the explanatory view of the CBC mode encryption operation (the 2) in the 1st embodiment.
[Drawing 16] It is the explanatory view of the CBC mode encryption operation (the 3) in the 1st embodiment.
[Drawing 17] It is the timing chart of the CBC mode encryption operation.
[Drawing 18] It is the explanatory view of the CBC mode decode operation (the 1) in the 1st embodiment.
[Drawing 19] It is the explanatory view of the CBC mode decode operation (the 2) in the 1st embodiment.
[Drawing 20] It is the explanatory view of the CBC mode decode operation (the 3) in the 1st embodiment.
[Drawing 21] It is the timing chart of the CBC mode decode operation.
[Drawing 22] It is the block diagram showing the composition of the cipher-processing circuit as the 2nd embodiment of this invention.
[Drawing 23] It is the figure showing comparison of the processing performed within one clock in the 1st embodiment and 2nd embodiment.
[Drawing 24] It is the basic explanatory view of the encryption processing in the CBC mode.
[Drawing 25] It is the basic explanatory view of encryption processing in CFB mode.
[Drawing 26] It is the figure showing the conventional parallel of the cipher-processing circuit in

the CBC mode.

[Drawing 27] It is the figure showing the conventional parallel of the cipher-processing circuit in CFB mode.

[Explanations of letters or numerals]

1 Data Storage Means

2 Function Processing Means

3 Mode Processing Means

4 1st Mode Processing Means

5 2nd Mode Processing Means

21, 28, 33, 36, 37, 44 Register

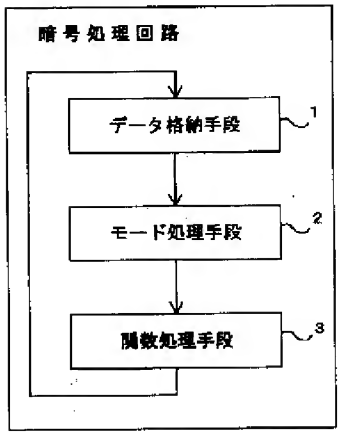
22, 23, 27, 31, 32, 35, 41, 46 Selector

24, 25, 30, 34, 42, 45 EXOR circuit

26 F Function (Processing Circuit)

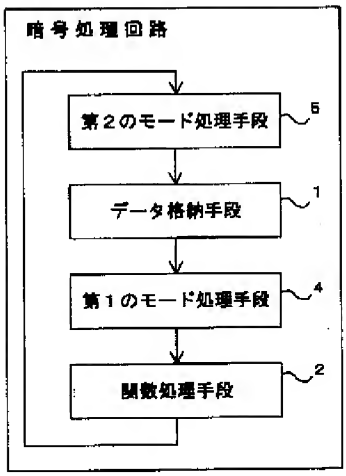
[Drawing 1]

第１の実施形態に対応する
原理構成ブロック図



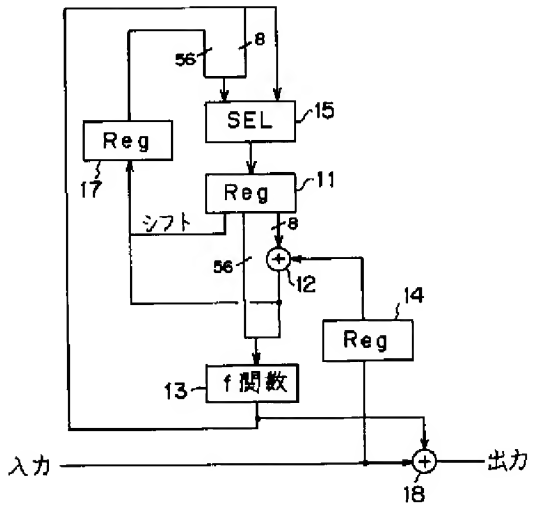
[Drawing 2]

第２の実施形態に対応する
原理構成ブロック図



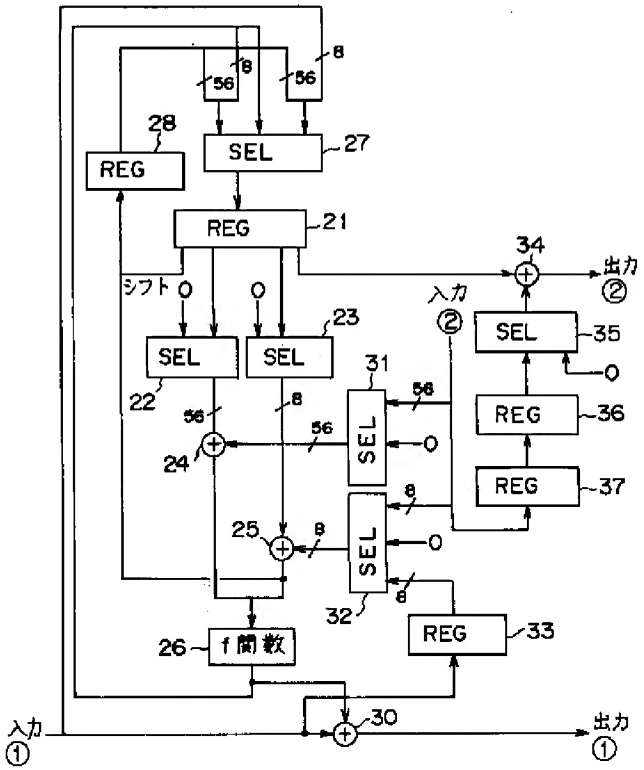
[Drawing 3]

本発明の第１の実施形態において使用される
CFBモード暗号化回路の構成を示すブロック図



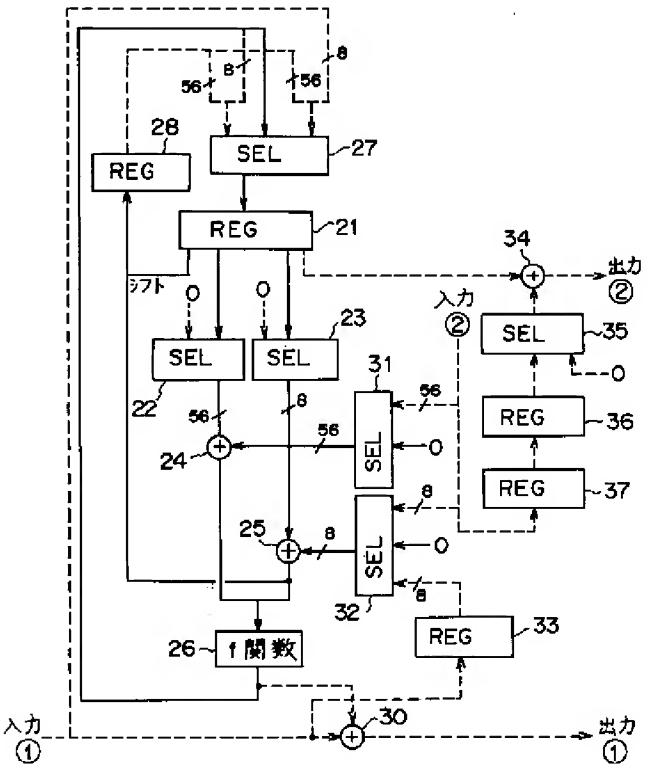
[Drawing 4]

本発明の第1の実施形態としての
暗号処理回路の構成を示すブロック図



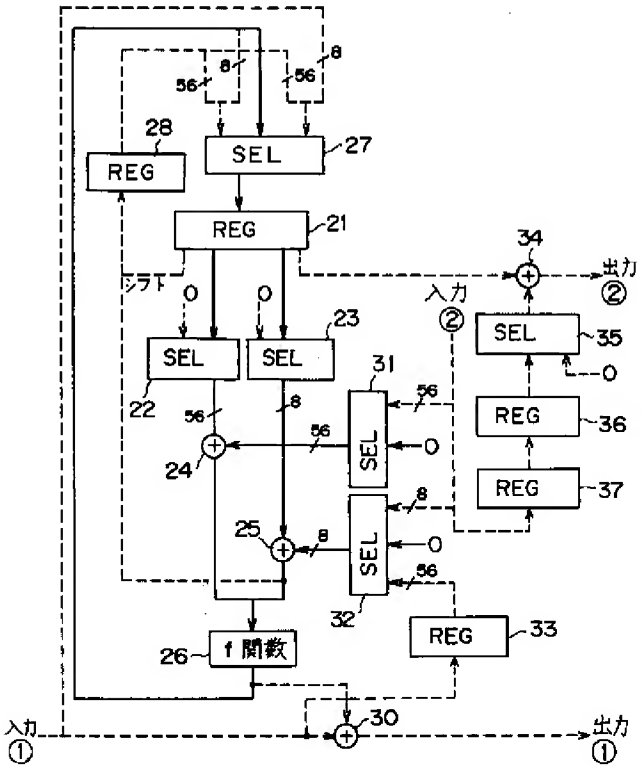
[Drawing 5]

第1の実施形態におけるCFBモード暗号化動作
(その1)の説明図



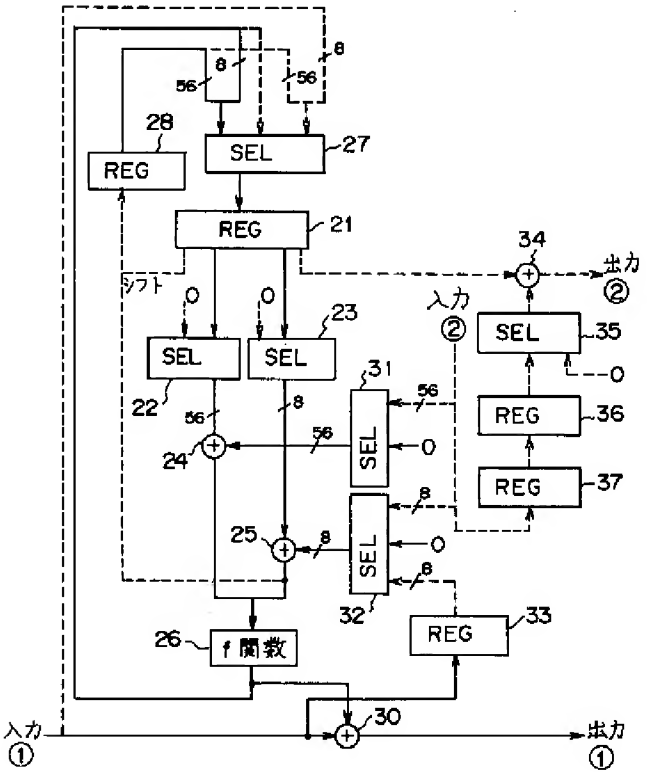
[Drawing 6]

第1の実施形態におけるCFBモード化暗号化動作
(その2)の説明図

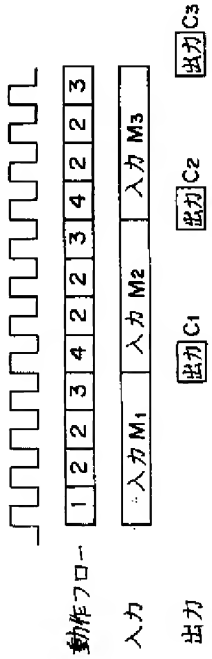


[Drawing 7]

第1の実施形態におけるCFBモード暗号化動作
(その3)の説明図

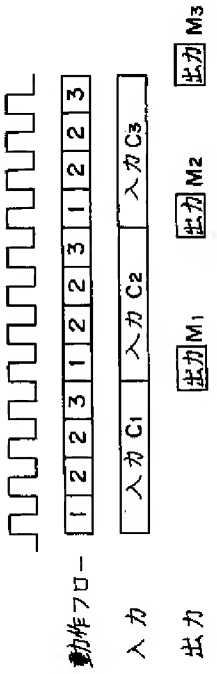


[Drawing 9]
CFBモード暗号化動作の
タイミングチャート



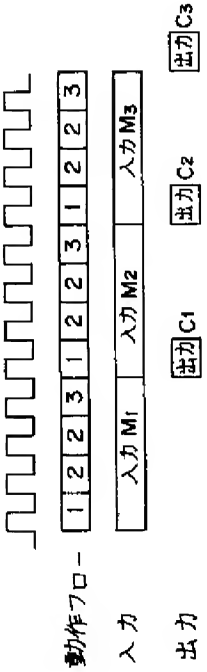
[Drawing 13]

CFBモード復号動作の
タイミングチャート



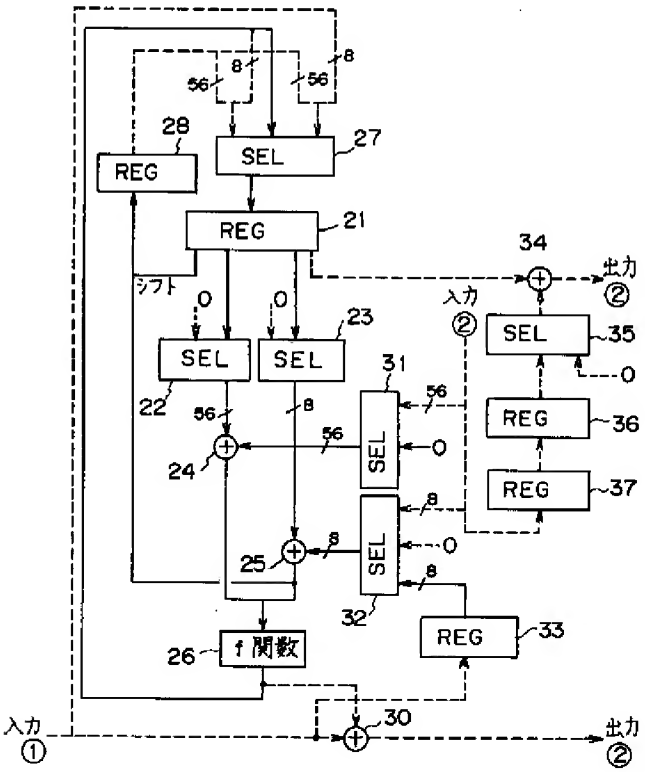
[Drawing 17]

CBCモード暗号化動作の
タイミングチャート



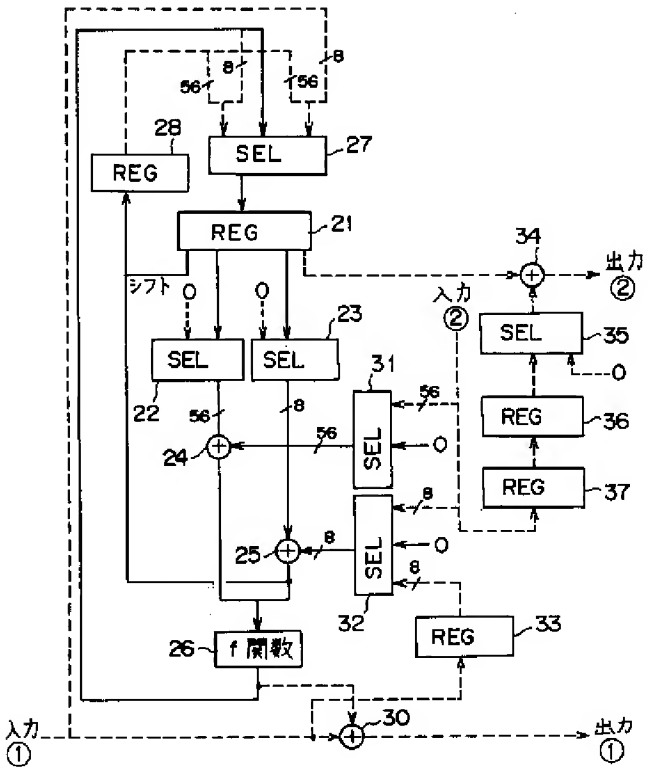
[Drawing 8]

第1の実施形態におけるCFBモード暗号化動作
（その4）の説明図

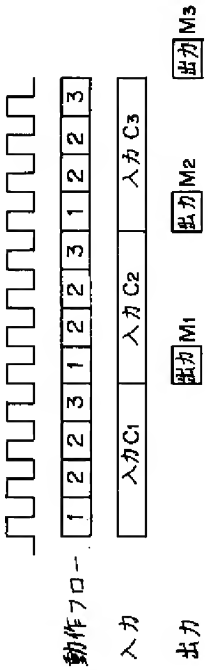


[Drawing 10]

第1の実施形態におけるCFBモード復号動作
(その1)の説明図

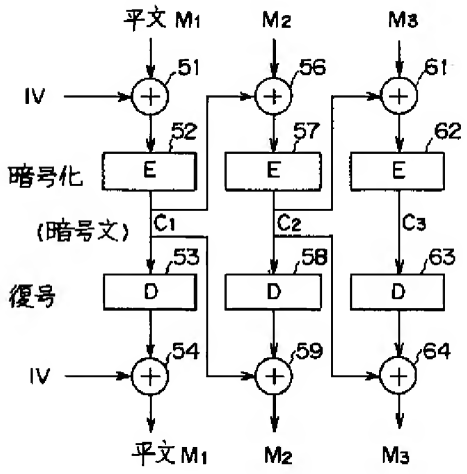


[Drawing 21]
CBCモード復号動作の
タイミングチャート



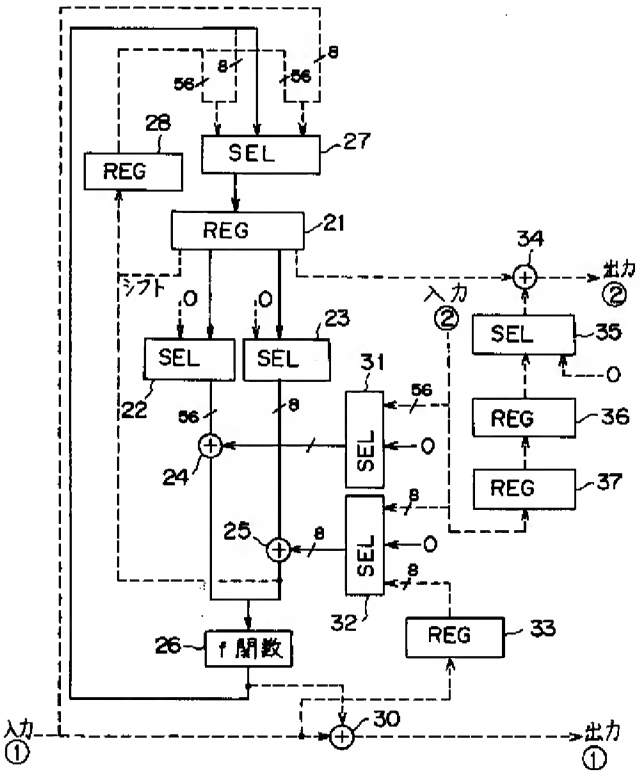
[Drawing 24]

CBCモードにおける暗号化処理の基本説明図



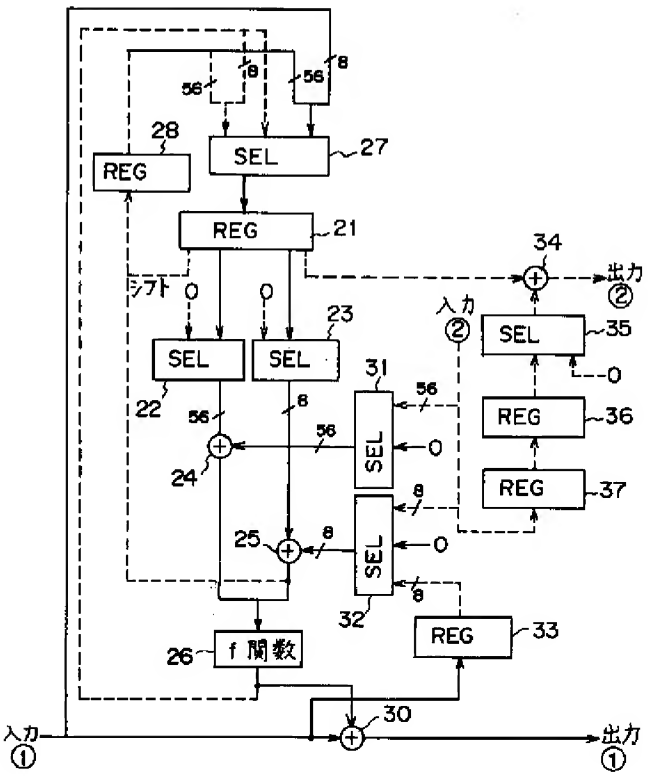
[Drawing 11]

第1の実施形態におけるCFBモード復号動作
(その2)の説明図



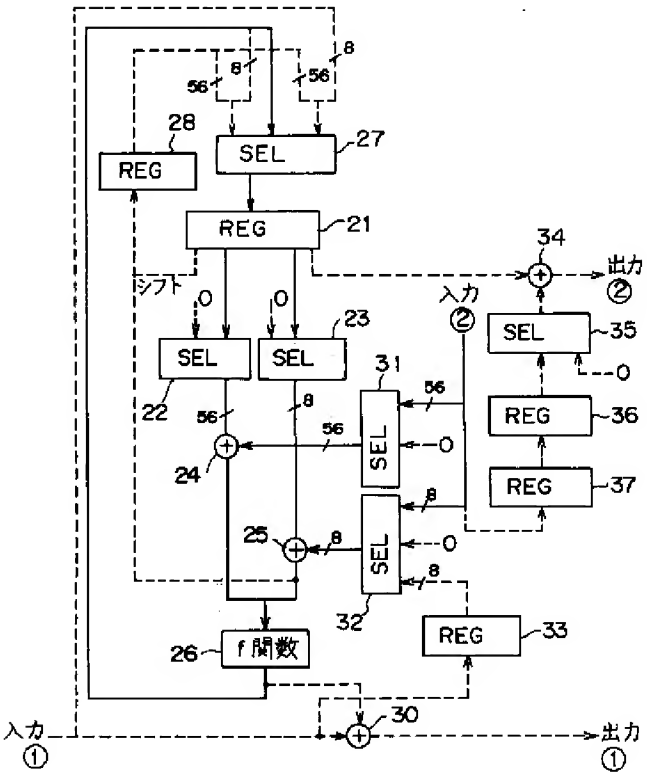
[Drawing 12]

第1の実施形態におけるCFBモード復号動作
(その3)の説明図



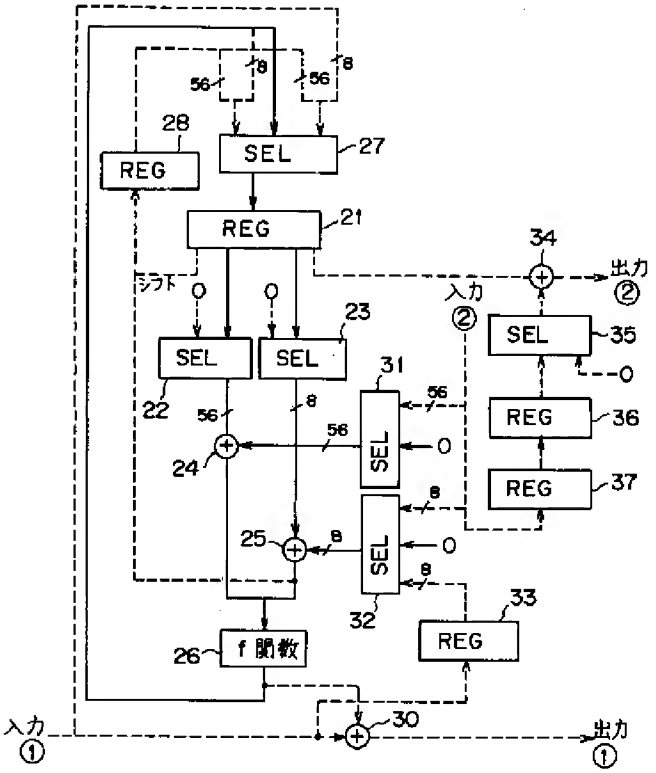
[Drawing 14]

第1の実施形態におけるCBCモード暗号化動作
(その1)の説明図



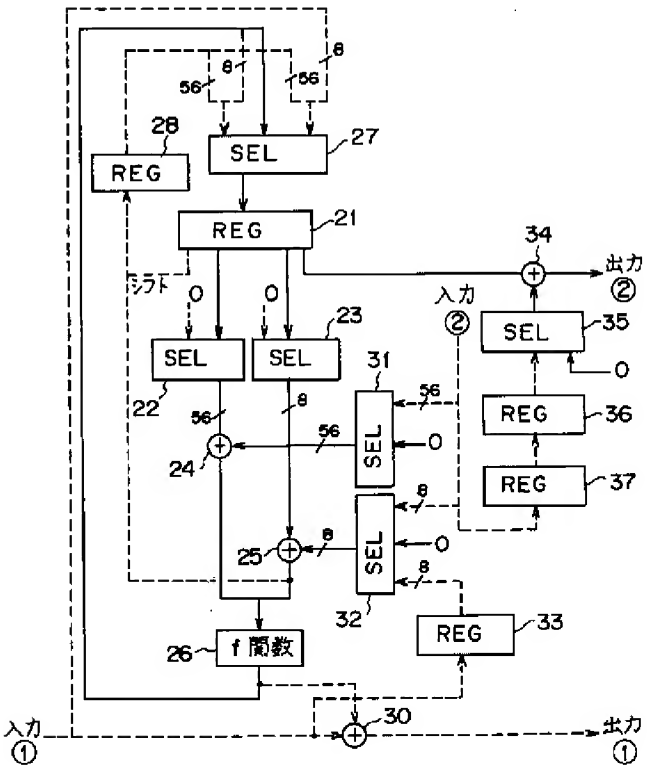
[Drawing 15]

第１の実施形態におけるＣＢＣモード暗号化動作
（その２）の説明図



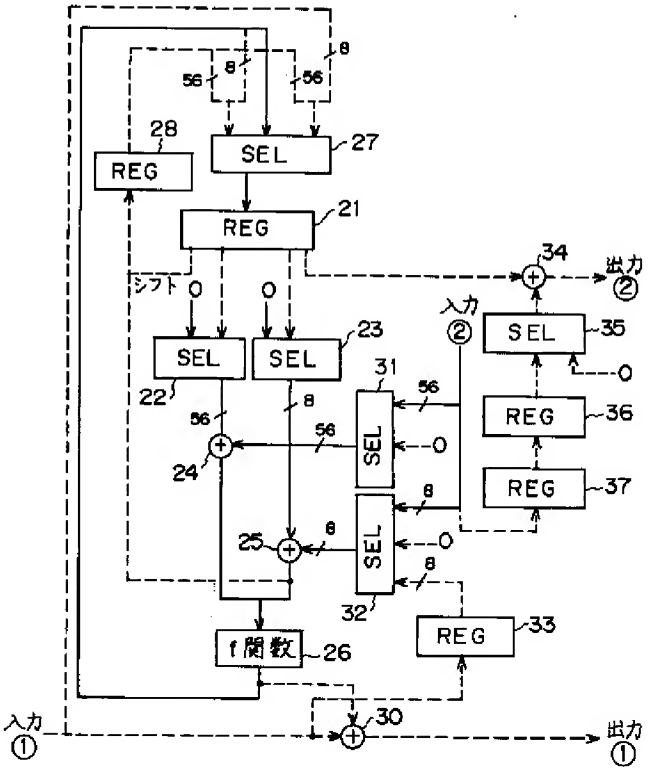
[Drawing 16]

第1の実施形態におけるCBCモード暗号化動作
(その3)の説明図



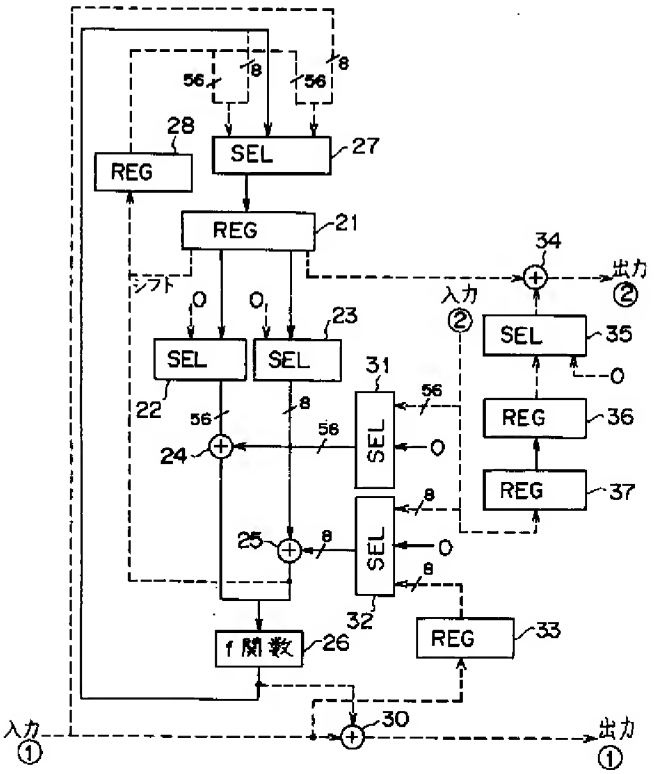
[Drawing 18]

第1の実施形態におけるCBCモード復号動作
(その1)の説明図



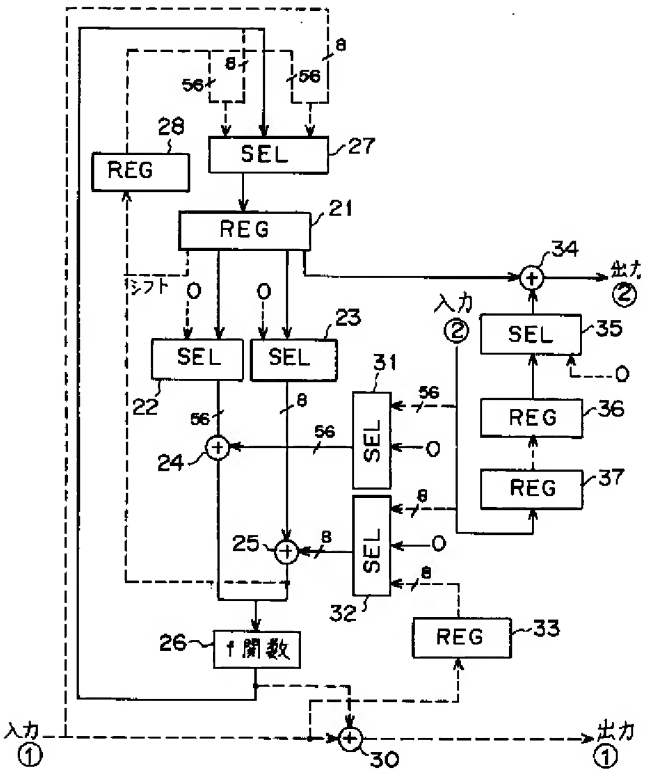
[Drawing 19]

第1の実施形態におけるCBCモード復号動作
(その2)の説明図



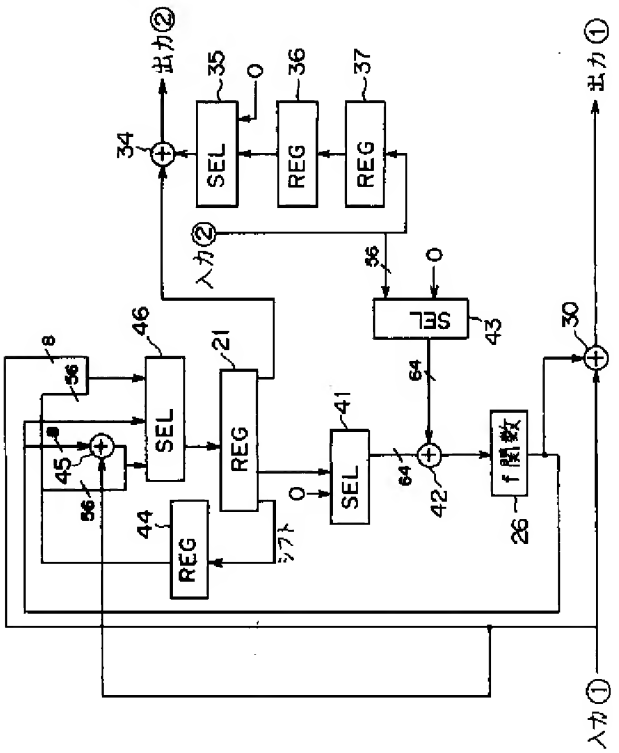
[Drawing 20]

第1の実施形態におけるCBCモード復号動作
(その3)の説明図



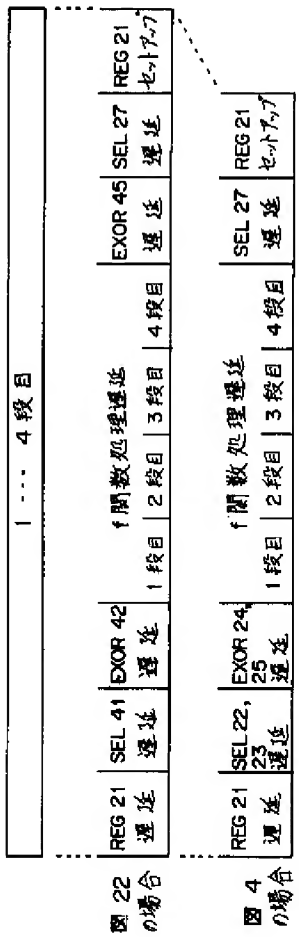
[Drawing 22]

本発明の第2の実施形態としての
暗号処理回路の構成を示すブロック図



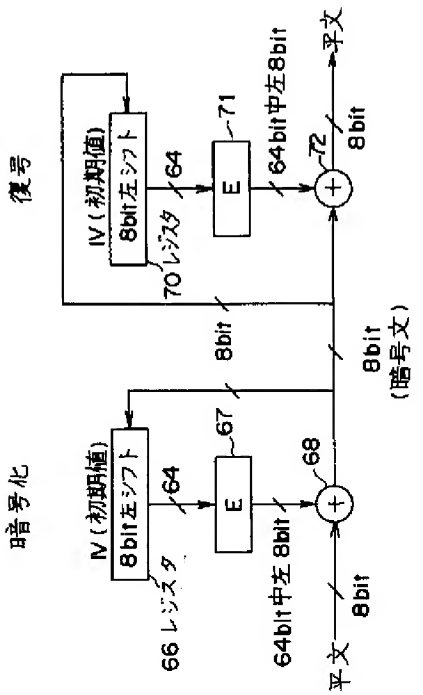
[Drawing 23]

第1の実施形態と第2の実施形態における
1つのクロック内で行われる処理の比較を示す図



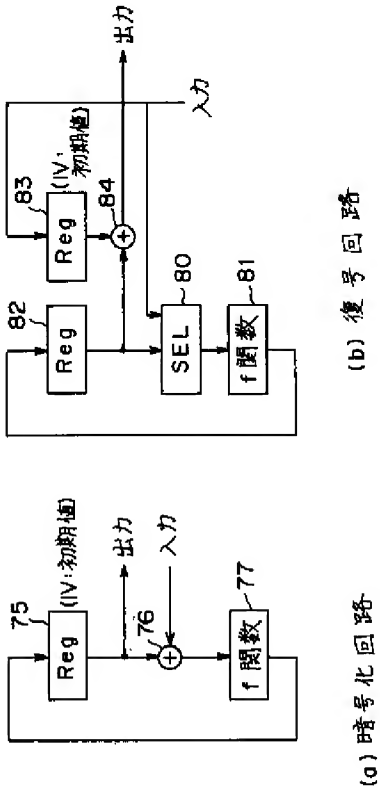
[Drawing 25]

CFBモードの暗号化処理の基本説明図



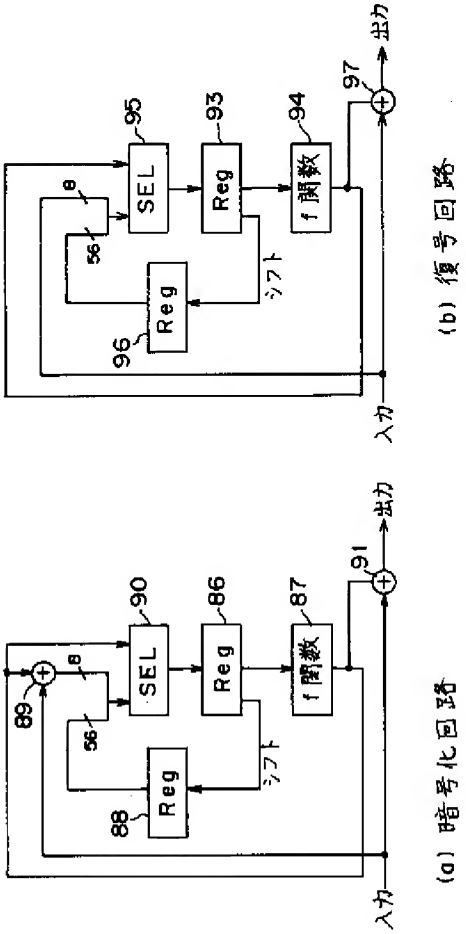
[Drawing 26]

CBCモードの暗号処理回路の従来例を示す図



[Drawing 27]

CFBモードの暗号処理回路の従来例を示す図



[Translation done.]